SBL-ONLINE:
IMPLEMENTING STUDIO-BASED LEARNING TECHNIQUES IN AN ONLINE INTRODUCTORY PROGRAMMING COURSE TO ADDRESS COMMON PROGRAMMING ERRORS AND MISCONCEPTIONS

A DISSERTATION SUBMITTED TO THE GRADUATE DIVISION OF THE UNIVERSITY OF HAWAI‘I AT MĀNOA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

COMMUNICATION AND INFORMATION SCIENCES

MAY 2013

By

Blanca J. Polo

Dissertation Committee:
Martha E. Crosby, Chairperson
Rich Gazan
Luz M. Quiroga
Scott Robertson
Marie Iding

UMI Number: 3572503

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

UMI 3572503

Published by ProQuest LLC (2013). Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.
All rights reserved. This work is protected against
unauthorized copying under Title 17, United States Code

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

© Copyright 2013
by
Blanca Judith Polo
All Rights Reserved

Dedication

I dedicate this dissertation to my wonderful family. Particularly to my loving and patient husband, Julio, who has gone above and beyond the call of duty. He nursed me when I was sick, helped me with his technical expertise and motivated me when I was feeling down. This accomplishment is as much his as it is mine.

Acknowledgements

First and foremost I want to thank my adviser and friend Martha Crosby. She believed in me when I was a master's student and provided me with the opportunity to become a teacher. When she saw that I enjoyed it, she encouraged me to pursue a PhD. She was my beacon through all the years it took to accomplish this endeavor. She stood right by me through the end.

I want to thank my mother Bertha for her support. She spent countless weekends babysitting my children so that I could study and do research. She made my life significantly better with her smile, support and loving care. I am very lucky to have her as my mother.

I want to thank my husband Julio who has helped me in so many ways. To write down what he has done would add another hundred pages to this document.

I want to thank my three girls: Bertha, Bianca and Isabella. They were very patient and understanding of the time I had to devote to my PhD. They grew up supporting me and encouraging me. Bertha is now a college graduate, Bianca is a sophomore in college and Isabella is eight years old and knows all too well what a dissertation is.

I want to thank my beloved brother Alberto and his wife Lluvia for letting me know that they believed in me. I appreciate that they always found a way to make me feel loved and supported even when oceans apart.

I want to thank my friends Monica Boon, Barbara Hotta, and Paula Alexandra Silva for their support in the last leg of this venture. Without them, this would have been significantly harder.

Abstract

Much research has been done in regards to student programming errors, online education and studio-based learning (SBL) in computer science education. This study furthers this area by bringing together this knowledge and applying it to proactively help students overcome impasses caused by common student programming errors.

This project proposes a pedagogy for proactive online teaching. Studio-Based Learning Online (SBL-Online) consists of three components, the SBL pedagogy, the strategic timing and content of the SBL sessions and the online nature of the implementation environment. The implementation of SBL-Online aims to provide a better quality of online computer programming courses promoting student-student interaction focused on the discussion of problematic curriculum issues.

Data collected prior to the application of SBL-Online provided counts of common student programming errors. This information was used to create the assignments that were later used during the SBL-Online sessions. In regards to its implementation, this project is two-fold; on the one hand it compares online vs. SBL incarnations of the same course. On the other hand it observes and analyzes what goes on during the SBL-Online sessions.

The data collection includes an end-of-semester questionnaire, a modified version of the motivated strategies for learning questionnaire (MSLQ), as well as thick data collection extracted from the course management system, student emails and SBL-Online session video recordings. Recordings' analyses of student interaction during SBL sessions play a very important role within this study.

The SBL-Online just as its face-to-face counterpart SBL, has shown the potential to help novice programmers overcome the limitations unique to introduction to programming courses. In most cases there is evidence on students' optimistic state of mind before, during and after SBL sessions highlighting the effects of this treatment. On the other hand there is evidence of renouncing and defeatist thoughts from the control group.

This research provides a unique insight into the inner workings and effects of SBL, including the identification of "learning moments," student-student interaction and the

building of camaraderie. SBL-Online impacted student's behavior, in different areas of the course, such as, attitude towards peers and attitude towards the course.

LIST OF TABLES

Table		Page
1	Assignment Turnout During the Fall 2008–Spring 2010 Period……	65
2	Summary of Errors During Midterm Exams …….…….…	66
3	ICS111 Programming Errors Listed in Descending Order ….	68
4	Session 1 Code Counts ……	100
5	Session 2 Code Counts ……	107
6	Session 3 Code Counts ……	113
7	Learning Moments Counts ……..	121
8	Code Counts Sorted by Count per Session …….	122
9	Code Categories Percentages ……..	123
10	Emails Sent to the Instructor ……...	127
11	Laulima Visits …….	128
12	MSLQ Sense of Connectedness and Community (Traditional) …….……	130
13	MSLQ Sense of Connectedness and Community (SBL_Online) …….…….	131
14	MSLQ Attitudes in Regards to Peer Learning …….	132
15	Comparison of SBL-Online vs Traditional Grades …….	141
16	Final Exam Comparisons ……..	142
E1	Assignment Grades ……..	364
E2	Quiz Grades …….	365
E3	Exam Grades ……...	366

LIST OF FIGURES

Figure		Page
1	SBL-Online	20
2	The SBL Process	25
3	The Scope of the SBL Project	27
4	Topics Causing Errors in CS1	38
5	ICS Majors and Non-Majors by Semester	45
6	Students by Campus Affiliation	46
7	Elluminate Reference Guide	51
8	Research Paradigm	55
9	Data Inventory	56
10	Coding Categories by Curtis and Lawson	59
11	Errors Related to Programming Tools and Task Understanding	61
12	General and Structural Programming Errors	61
13	Errors Specific to the Java Programming Language	63
14	Assignment Delivery Setbacks	64
15	The Most Common ICS111 Errors and the Lecture Where the Related Material Is First Introduced	69
16	Lectures That Cover Troublesome Material in Chronological Order	70
17	Semester Chronology of Events Based on Broadcast Sessions, Part 1	74
18	Semester Chronology of Events Based on Broadcast Sessions, Part 2	75
19	Semester Chronology of Events Based on Broadcast Sessions, Part 3	76
20	Data Collection Schedule	84

21	SBL-Online Team Participants ……...…….	86
22	The Initial Video Transcription Form ...……	87
23	The Final Video Transcription Form ...…….	88
24	Transcription Code Used for This Study……...……	89
25	Discrete Chat Pieces Sharing One Code (Session 1, Team 4, 29:53) ..…….…….…….	94
26	Example of Chat and Talk at the Same Time (Session 3, Team 1, 6:16)…….	95
27	Temporarily Becoming an Author (Session 1, Team 3, 57:28).	96
28	FBG vs. HYP (Session 1, Team 3 19:48) ……...……	97
29	Error in the Code ……..……..	102
30	Chat/Talk Interaction Discussion ……..	103
31	Transcript Excerpt 1 ……..	109
32	Transcript Excerpt 2 ……..	110
33	Transcript Excerpt 3 ……..	114
34	Transcript Excerpt 4 ……..	118
35	Becoming Efficient (Session 2, Team 1, 21:43) …….	120
36	Attitudes Towards the Class ……..	134
37	What Impacted Learning……...	135
38	The Impact of Feedback in Students …….	136
39	Social Interaction ……..	150
B1	Coding Scheme to Describe Utterances in Online Collaboration …….	180
C1	Slot Machine GUI ……..	185
D1	Session 1–Team 1 Transcript ……	198
D2	Session 1–Team 2 Transcript ……	224
D3	Session 1–Team 3 Transcript ……	240
D4	Session 1–Team 4 Transcript ……	262
D5	Session 2–Team 1 Transcript ……	278
D6	Session 2–Team 2 Transcript ……	295

D7	Session 2–Team 3 Transcript ……	308
D8	Session 3–Team 1 Transcript ……	315
D9	Session 3–Team 2 Transcript ……	329
D10	Session 3–Team 3 Transcript ……	348
E1	Assignment Submission Messages: Control Group ……..	367
E2	Assignment Submission Messages: SBL-Online Session 1…..	368
E3	Assignment Submission Messages: SBL-Online Session 2…..	369
E4	Assignment Submission Message: SBL-Online Session 3….....	370

Table of Contents

Dedication iii

Acknowledgements iv

Abstract v

List of Tables vii

List of Figures viii

Chapter 1. Introduction 1

- **Studio-Based-Learning (SBL) Defined** 1
- **SBL-Online, a Good Fit for Online CS1** 2
- **The Importance of CS1** 2
- **Problems Characteristic of CS1** 3
- **Reasons for Selecting SBL-Online as a Viable Solution for CS1 Problems** 3
- **Introducing the Research Questions** 4
- **In Preparation for This Study** 5
- **Terminology** 5
- **External Influences Regarding This Project** 6
- **The Scope of This Dissertation** 6
- **Overview of This Document** 6
- **Summary** 7

Chapter 2. Theoretical Background 8

- **The Purpose of Education** 8
- **The Notion of Online Education** 9
- **The Importance of Interaction in Online Education** 9
- **Teamwork as Means for Student Interaction** 10
- **The Role of Social Constructivism in the Classroom** 11
- **The Implementation of Constructivist Learning Environments** 11
- **The Role of Computer-Mediated Communication as Support of Online Constructivist Environments** 12
- **The Differences Between Studio-Based Learning Online and Teamwork** 13
- **The Essentials of SBL** 13

Goals of SBL Implementations in the Computer Science Classroom.....14
New Perspectives on an Old Curriculum 15
Outcomes of SBL Implementations in Computer Science Education 17
How SBL-Online Differs from Other SBL Implementations.....18
Summary 19

Chapter 3. SBL-Online Revisited.....20
The Purpose of This Study.....21
How SBL Became Part of the Computer Science Curriculum.....22
SBL Components in Programming Courses 23
The SBL Design Crit 24
Effects of SBL on Students.....25
SBL Implementations.....26
SBL-Based Degrees.....27
SBL Courses 27
SBL Essentials.....28
The Benefits of SBL 28
Motivations for Implementing SBL.....29
Problems Derived from SBL 30
Online Learning Overview.....31
Distance Education 31
The Influence of Technology.....32
The Evolution of CS1.....32
Debates About CS1 Pedagogies 33
Reasons for High Attrition in CS Programs 33
Perspectives on Programming Errors 35
CS1 Error Categorizations.....36
Identifying Clusters of Recurrent Errors in CS1 39
SBL-Online Defined.....40
Preliminary Outcomes.....40
Summary 41

Chapter 4. Methodology....42
Introduction....42
SBL-Online Research Questions....42
SBL Project Overview....43
SBL Project Research Questions and Their Relationship to SBL-Online....43
Population and Sample Description....44
Participants....46
Settings for This Study....47
Laulima, the University of Hawai‘i Course Management System....47
Course Components....47
Laulima Integrated Course Components....48
Laulima Add-On Components....50
Components External to Laulima....52
Institution Review Board (IRB)....53
The Research Paradigm Defined....54
Procedures for Creating, Coding, and Analyzing Data....55
Instrumentation....57
The Motivated Strategies for Learning Questionnaire (MSLQ)....57
End-of-Semester Questionnaire....58
Inside SBL-Online....58
The Proactive Instructor....60
Collecting Grades....60
Error Categorization and Descriptions....61
Errors as a Guiding Light....64
Building and Placing SBL-Online Sessions....69
SBL Project Procedures Within SBL-Online....73
Describing the SBL-Online Studio Session....76
During the SBL-Online....78
Reflections About Validity and Reliability....79
Limitations of This Study....79
Summary....80

Chapter 5. The Presentation of Data 81
Introduction 81
Design and Methodology of This Study 81
The Purpose of This Study 81
Research Questions, and How They Are Addressed 82
Data Creation and Collection 83
SBL-Online Course Elements Analyzed 85
Video Transcription Process 87
Video Transcript Format 87
The Process of Adapting a Code to Describe SBL-Online Interaction 88
Codes and Categories 90
Coding in Context 93
The Chat and the Spoken Word 93
Code Selection Process 95
Video Analysis Findings 98
Inside SBL-Online 98
First Round of SBL-Online Sessions 99
Second Round of SBL: Online Sessions 107
Third Round of SBL-Online Sessions 113
Sessions Overview 119
Becoming Efficient 119
Code Counts' Analysis 120
Student-Student Interaction During the Traditional Implementation 124
Comparing SBL-Online vs. Traditional Implementations 127
Common Elements Shared By This Study and the SBL Project 128
Modified MSLQ 129
End of Semester Questionnaire 133
Shared Course Elements 141
Student Performance 141
Reflections 142
Summary 142

Chapter 6. Conclusions and Future Work 144
Addressing the Research Questions 144
RQ1. Is there any indication how the application of SBL-Online aids a beginning programming class? 144
RQ2. What are the interaction highlights? 145
RQ3. Is there any indication that learning is taking place? 146
RQ4. Are students' misconceptions being identified and addressed? 147
RQ5. Are students acquiring knowledge that will transfer into other course activities? 148
Unique Characteristics of SBL-Online 148
Student-Student Interaction 149
Social Interaction During SBL-Online 150
Peer Reviews 150
Errors 151
Assignment Submission Messages 152
SBL-Online to Help a Beginning Programming Class 152
Contributions Made by This Study 153
The Formal Conceptualization of Proactive Teaching 153
The Implementation of SBL Pedagogies in a Fully Online Course 153
The Adaptation of a Transcript Code to Address SBL-Online Interaction 154
The Identification of Patterns That May Indicate a "Learning Moment" 154
An Inner View into the Mechanics of SBL Interaction 154
Guidelines to a Successful SBL-Online Implementation 155
Reflections About This Study 156
Conclusions 157

Appendix A. Assessments 158
Midterm Exam 1 158
Midterm Exam 2 163
Final Exam 166

Appendix B. Instruments for Gathering Data 169

Consent Form **169**
Modified MSLQ: Motivated Strategies for Learning Questionnaire **172**
Motivation Set **172**
Task Value 172
Self Efficacy 172
Learning Strategies Set **172**
Critical Thinking 172
Peer Learning 173
Performance Set **174**
Task Goal Orientation 174
Intrinsic Goal 174
Classroom Community Set **175**
Sense of Communities Connectedness 175
Sense of Communities Learning 175
End-of-Semester Questionnaire **176**
Coding Scheme to Describe Utterances in Online Collaboration **180**
Appendix C. Assignments and Inspection Forms **181**
Assignment 6 - Calculating Body Mass Index **181**
BMI Calculation Instructions 181
Assignment 8 - Slot Machine Applet **184**
Slot Machine Instructions 184
Assignment 12 - CD Store **187**
CDStore Instructions 187
Assignment 10 - CD Object with Exception **190**
CD Object with Exception Instructions 190
Inspection Forms **196**
Appendix D. Session Transcripts and Learning Moments Script **198**
Transcripts **198**
Perl Script to Count Learning Moments **362**

Appendix E. Semester Grades and Submission Messages …….……..364

Glossary of Terms …….……..……..……...371

References ……..……..……..……...375

Chapter 1. Introduction

Studio-Based-Learning (SBL) Defined

Studio-Based Learning (SBL) is an "inquiry apprenticeship model that follows problem-based learning but allows a more pervasive person-centered approach" (Brocato, 2009). This approach is modeled after the architectural design studio. It is founded on the concepts of reflection and social constructivism (Hundhausen, Agrawal, Fairbrother, & Trevisan, 2010; Lynch, Carbone, Arnott, & Jamieson, 2002). The key elements in studio-based learning are collaborative learning, interactive problem solving, public presentation and review by peers in small groups (Docherty, Sutton, Brereton, Kaplan, & Brown, 2000; Woodley & Kamin, 2007). In studio-based learning, learning is seen as "a process in which one constructs one's own understandings through active engagement with one's environment, especially through constructing and exploring one's own artifacts" (Hundhausen, Narayanan, & Crosby, 2008).

Examples of the Studio-Based Learning (SBL) teaching method may be found in schools of architecture, where students undertake a design project under the supervision of their instructor. In this pedagogy, the instructor is not the focus of the class; students instead, work together to solve problems and answer each other's questions.

SBL may be implemented in a variety of way however it has characteristic elements that remain constant:

- Students are given a non-trivial problem to solve, independently or in teams.
- Students present their solutions to other students or teams.
- Students receive feedback from their peers.
- Students incorporate feedback received into their solutions.

Similar process are described by Hendrix, Myneni, Narayanan, and Ross (2010), Hundhausen, Agrawal, Fairbrother, and Trevisan (2010), and Pat Donahue (Bray, 2012). It is argued that SBL pedagogies promote student interaction and promote holistic skills including critical and creative thinking, inquiry and reflection through frequent peer review and critiquing.

SBL-Online, a Good Fit for Online CS1

As evidenced by the multiple implementations of SBL in computer science education, it can be argued that SBL pedagogies are a good fit for computer science students. SBL-Online, as described and implemented in this study provides students with a platform for student-student interaction where controversial topics and recurrent errors can be discussed and addressed. SBL-Online is different to previous implementations, first and foremost because it is implemented fully online. Secondly, SBL-Online requires the creation of special SBL assignments based on recurrent errors from past CS1 students. Lastly Online-SBL sessions are placed at key points in the curriculum where concentrations of errors have shown to occur.

SBL-Online was implemented in CS1. CS1 is a cornerstone course in computer science education. In order to better understand the impact of SBL-Online in computer science education, it becomes necessary to learn more about CS1.

The Importance of CS1

In order to carry out this study, the CS1 course was selected because of its importance as the de-facto introductory programming course as described by the Association for Computing Machinery (ACM) in pages 22 to 24 of the Computing Curricula 2001 Computer Science Final Report (Chang, 2001) several institutions offer a two-course sequence, CS1 and CS2. This sequence even while covering the same basic material has evolved due to constant changes in technology and multiple pedagogical approaches. CS1 is the "Introduction to Computer Science 1" course, known as ICS111, in several of the University of Hawai‘i campuses. The course used for this study is an online version of ICS111 using the Java programming language. This study seeks an innovative and effective way to deliver CS1 material.

Problems Characteristic of CS1

Difficulties in learning programming often result in lack of motivation and high dropout and failure rates. This affects not just introductory programming courses but also the computer science degree (Hundhausen et al., 2008; Simon, Kinnunen, Porter, & Zazkis, 2010). Finding the precise pedagogy and correct tools to teach programming has been a goal for computer science teachers for many years (Evans, 1996; Ficocelli & Gregg, 2011; Gross & Powers, 2005).

In order to help students, it is important to provide assistance when students need it the most. Whittington and Bills (2004) suggest identifying and addressing curriculum course problem areas caused by difficult concepts or techniques that inhibit student learning. Unfortunately, these problems usually surface after a failed test or assignment. Grading students' projects, tests and quizzes is a way to identify troublesome concepts that hinder learning but those cues surface too late. In a traditional face-to-face course, after finding multiple common errors it is possible to dedicate time for corrective action in an attempt to patch student's knowledge. However, in some online situations, the recurrence of errors may discourage students to continue pursuing a degree in computer science. A new strategy is required to solve this problem, and that strategy may be the application of SBL-Online as proposed by this study.

Reasons for Selecting SBL-Online as a Viable Solution for CS1 Problems

Many studies have been done on how to prevent CS1 students from dropping out. Discouraged students often times drop not only the course, but drop out of the major altogether (Hundhausen et al., 2008). The SBL Project longitudinal studies have shown higher retention rates in computer science students who experience SBL (www.studiobasedlearning.org). Since student-student interaction is a powerful predictor of students' intentions to persist in the major (Barker, McDowell, & Kalahar, 2009; Moore, 1989), promoting interaction throughout the semester by implementing of SBL-Online, should increase retention as well.

SBL-Online may not only help prevent student dropout but may also address one of the regular challenges in programming classes. Students usually only see a final working version of programming code during lectures as well as in books. While lectures may address how the code came to be, books show finished code without addressing the thought process that took place to create it. However, explaining and learning the thought process invested into product creation may be fundamental for learning. The SBL pedagogy provides opportunities to learn how programs are designed and built.

SBL-Online may provide students with learning benefits, and it may also help them feel comfortable participating remotely with their peers. Nowadays, students are expected to be able to communicate and deliver in a world where computer communication is pervasive. Online instruction has become commonplace and globally distributed teams must work together (Lee & Choi, 2011). The application of SBL techniques to online courses is highly relevant because online course offerings have been on the rise, and nearly tripled between 1993 and 2003. Most public institutions acknowledge online instruction as a critical part of their long-term plans (Crawford-Ferre & Wiest, 2012).

Introducing the Research Questions

The purpose of this study is to describe how SBL-Online impacts student's behavior, attitude towards the course and semester grades in an online CS1 course. This study is addressed in parts, as described by the following research questions:

Is there any indication of how the application of SBL-Online aids a beginning programming class?

During the SBL-Online sessions,

- What are the interaction highlights?
- Can indicators of learning be documented?
- Are students' misconceptions being identified?
- Are students' misconceptions addressed?
- Are students acquiring knowledge that will transfer to other course activities?

In Preparation for This Study

In order to prepare this study, the CS1 course, known at Leeward Community College as ICS111, was implemented in the exact same way for four semesters preceding this study. This was true for the course components as well as their timings. Course components included recorded lectures, assignments, tests and quizzes. While lectures were broadcasted as well as available online, other course components were available via a course management system. Availability of course components was timed according to the lecture broadcast schedule.

The homogeneity of course delivery allowed for the collection of student performance data across these four semesters. Data collected was used to identify the topics where students made the most mistakes. Programming and conceptual errors were identified, classified, quantified, and then used as the base for the creation of assignments that would become part of the SBL-Online sessions. The SBL-Online sessions took place right after the broadcast of the lecture, in which the identified troublesome material was introduced.

SBL-Online sessions are intended to force students to use programming constructs as well as concepts that have shown to be challenging to students in the past. SBL-Online is expected to expose students to the thinking process of others that are attempting to learn the same skills. SBL-Online may also help students learn to accept critique of their work, not just from the instructor but also from their peers.

Terminology

For the remainder of this document the use of the term “CS1” will refer to Computer Science courses within other universities, while CS1 courses offered at Leeward Community College will be referred to as ICS111.

External Influences Regarding This Project

This study is part of a project of simultaneous SBL implementations called The SBL Project, funded by the National Science Foundation. These SBL implementations were done by a group of professors hosted at different institutions, mainly Washington State University, Auburn University and the University of Hawai‘i at Mānoa. Preliminary results of this research group show higher retention, and sometimes, higher grades for students that have experienced SBL. From this research group, the project that most closely resembles this study is the Online Studio-Based Learning Environment (OSBLE; Hundhausen, Agrawal, Fairbrother, & Trevisan, 2010). However, OSBLE has only been used by students in face-to-face courses.

Despite numerous SBL implementations, nobody has recorded sessions in the manner done by this study, and no SBL implementation has yet been done in an online CS1 class.

The Scope of This Dissertation

This study does not intend to address issues related to online instruction as a valid or effective pedagogy, nor will it evaluate the impact of computer-mediated communication (CMC) in the studio-based learning pedagogy. This study will not compare online tools, nor will it compare face-to-face SBL implementations with SBL-Online. This study does not consider how different levels of computing literacy affect the learning outcomes of CS1 students.

Overview of This Document

This chapter gives an overview of this study. It describes the differences between SBL-Online and other SBL implementations. It explains the limitations and delimitations of the study.

Chapter 2 presents the theoretical background and frames this research into the field of educations, more specifically, computer science education.

Chapter 3 contains the literature review that gives the theoretical background for this study. It includes listings and reviews of SBL implementations as well as studies about common programming errors.

Chapter 4 includes the methodology for this study including the description of the data collection needed in order to apply this methodology. It describes the overall design of this study, including the description of procedures for the data collection and analysis, as well as the technologies involved in this study.

Chapter 5 focuses on the presentation and analysis of the data collected. It reveals the inner workings of SBL-Online and its impact on students.

Chapter 6 presents the conclusions of this study, a postmortem on the process and the expectations of future research.

Summary

This chapter presented an overview about current problems in CS1. CS1 is the cornerstone course in almost every computer science degree. CS1 is a crucial course, as important as it is difficult for students. This chapter presented the SBL pedagogy as a viable solution for addressing CS1 problems. The research questions that guide this study supporting the application of SBL-Online were presented.

Chapter 2. Theoretical Background

Chapter 1 introduced the objectives and research questions of this dissertation. This chapter will present the theoretical background and the situation of this research within the fields of computer science education and online education. It will begin by addressing the purpose and evolution of education from the traditional model into pedagogies based on social constructivism. It will continue describing the growth of online education and how social constructivism can has been implemented within this evolving paradigm. Based on problems specific to the computer science education domain, studio-based learning pedagogies emerge as a constructivist approach that may help solve these problems, while SBL-Online may address similar issues in online computer science education.

The Purpose of Education

According to Hanson et al. (1997), the purpose of education is to transmit information and skills to students. Books are the main representations of knowledge, while teachers are the means to connect students with this body of knowledge. In this traditional model of learning, the academic controls the flow of information and students are considered separate entities (Lynch & Markham, 2003).

Although this traditional model of learning has been used for many years, research has shown that there are alternative effective methods of teaching. Education has evolved incorporating other elements that influence it. According to Sumner (2000, p. 272), "While much of our traditional understanding about learning sees it as an individualized process, there is a growing body of literature that addresses the importance of social learning." The instructor is no longer perceived as the only way to bridge the gap between the body of knowledge and the student. Nowadays this gap can be bridged with the incorporation of multimedia technology.

The Notion of Online Education

In 1977, Holmberg defined distance education as

> the various forms of study at all levels which are not under the continuous, immediate supervision of tutors present with their students in lecture rooms or on the same premises, but which, nevertheless, benefit from the planning, guidance and tuition of a tutorial organization. (Holmberg, 1977, p. 9)

Later in 1997, Hillary Perraton (as quoted by Hanson et al., 1997, p. 7) defined distance education as "an educational process in which a significant proportion of the teaching is conducted by someone removed in space or time from the learner." In 2010 Schlosser and Simonson (2010, p. 1) define distance education as "[i]nstitution-based, formal education where the learning group is separated, and where interactive telecommunication systems are used to connect learners, resources and instructors."

The above-cited definitions present broad generalizations and we can see that the terms have evolved from "distance education" where didactic material was delivered by correspondence at the end of the 19th century to "online education," emphasizing the use of the Internet to deliver multimedia content at the beginning or the 21st century (Sumner, 2000, p. 273).

Education evolved from its traditional form, where the teacher lectured and students listened towards the implementation of different pedagogies inspired by social constructivism. The importance of socialization in learning, specifically student-student interaction has driven many studies and observations based on the perceived potential for learning that takes place within (Laurillard, 2002, p. 159).

The Importance of Interaction in Online Education

Moore (1989) studied the importance of interaction in online education. He advocates that three kinds of interaction are needed in distance education in order to provide students with a more complete learning environment. The three types of interaction as described by Moore are:

1. Learner-content interaction encompasses actions taken by students to directly learn from the course material. The process of creating content is vital because the content needs to facilitate the learner's understanding of the course material.

2. Learner-instructor interaction represents the dialog between the student and the teacher. Moore emphasizes the importance of this type of interaction to motivate students to learn. This kind of interaction has more influence in student learning than learner-content interaction.

3. Learner-learner interaction consists in the communication between learners. It is illustrated by the exchange of ideas between the students. This interaction takes place with or without the instructor's presence.

Moore also argues that if distance education employs only one medium then it is probable that only one kind of interaction is permitted or properly applied. Different kinds of interaction call for different media, which can be provided with current Internet technologies.

Teamwork as Means for Student Interaction

Online instruction should go beyond educational material delivery and allow for student-student interaction. One such an interaction can be achieved by assigning teamwork. "Teamwork" is defined by the Merriam-Webster online dictionary as: "work done by several associates with each doing a part but all subordinating personal prominence to the efficiency of the whole" (http://www.merriam-webster.com/dictionary/teamwork). In comparing face-to-face team work with online team work, Qiu, Tay, and Wu (2009) found online teamwork to be more task oriented, while face-to-face team work was more cohesive and personal. On the other hand Bhagyavati, Kurkovsky, and Whitehead, 2005 decided to enhance student's traditional classroom experience by providing a forum to support asynchronous discussions. This action resulted in successful learning outcomes and student satisfaction. Providing the means for interactivity in the classroom has the potential to enhance student experience.

The Role of Social Constructivism in the Classroom

Pedagogies that promote student-student interaction fall under the umbrella of constructivism. Constructivism provides an alternative to traditional education. There is no specific constructivist theory of learning, instead there are characteristics exhibited by constructivist pedagogies. Constructivism posits the facts that the learner constructs knowledge based on prior knowledge and that constructivism is socially mediated (Woolfolk, 2013).

Constructivists assume that the collaborative building of meaning and understanding takes place as learners try to make sense of their experiences (Woolfolk, 2013). This knowledge goes beyond the transfer of information from person to person. This process is therefore more than a student reaction to instruction, it allows the learner to interact with the environment to build meaning and to acquire knowledge (Jonassen, Davidson, Collins, Campbell, & Haag, 1995).

Constructivist educators strive to engage learners in such a way that knowledge can be transferred into new situations (Jonassen et al., 1995). With this in mind, constructivists consider that students should be encouraged to solve complex problems that contain multiple elements and can be solved in a variety of ways (Woolfolk, 2013). These projects should focus on solving real world relevant problems (Jonassen, 1999), and would be better addressed from a constructivist learning environment (CLE).

The Implementation of Constructivist Learning Environments

According to Jonassen and Rohrer-Murphy (1999, p. 69), "a constructivist learning environment (CLE) has several components: a problem-project space, related cases, information resources, cognitive tools and conversation and collaboration tools." Based on these principles, the literature reports on a number of ways in which social constructivism can be applied. These include face-to-face as well as online implementations. These include contributing student pedagogies (CSP), computer mediated communication (CMC), computer supported collaborative work (CSCW), problem-based learning (PBL) and studio-based learning (SBL) among others.

Contributing student pedagogies (CSP) encourage students to contribute to the learning of others and to value the contributions of others (Hamer et al., 2008). CSP encourages contributions to the learning of others in a variety of ways such as providing demonstrations, tutorials, examples or giving feedback (Hamer, Luxton-Reilly, Purchase, & Sheard, 2011). CSP encompasses the methodologies of many pedagogies. However, its main characteristic is that it promotes the transfer of knowledge from student to student rather than from the teacher to the student.

Knowledge transfer and collaboration may take place online as it does face-to-face. However, when interacting online, students rely on digital tools. Tools for online collaboration are varied and afford different actions. One such action is the ability of students to initiate communication at anytime. This possibility is perceived by students as possessing control over the learning environment (Moore, 1989). To be able to interact with others at will provides positive social interdependence and allows students to exchange resources, provide and receive feedback and help as well as to encourage each other to succeed (Curtis & Lawson, 2001). Bruckman (2002, p. 61) describes learning as a social process supported by the Internet, while Peters (2000, p. 16) affirms that "the digital learning environment will probably be the most efficacious enabler of independent and self-determined learning."

The Role of Computer-Mediated Communication as Support of Online Constructivist Environments

Based in the digital world and supporting the principles of social constructivism, stands computer-mediated communication. Wolz et al. (1997, p. 51) define computer-mediated communication (CMC) as "any form of interpersonal communication that uses some form of computer technology to transmit, store, annotate, or present information that has been created by one or more participants." Tools that support CMC include two-way technologies such as email, video and audio conferencing, chat, as well as groupware among others. In CMC student-student interaction and collaboration are considered central to student learning. Knowledge is the product of interaction among students (Laurillard, 2002).

Online collaboration is supported by available technologies such as Google and Skype. Course management systems such as blackboard and moodle support online education and provide a converging point for the distribution of course material. The University of Hawai‘i counts with its own course management system called “Laulima,” which is based in Sakai. A mix of tools similar to the ones mentioned allow for online teamwork to take place.

Students willing to work in teams and available technologies are not enough to successfully promote collaboration and the transfer of knowledge in an online course. The need for a more structured collaborative approach that will incorporate the constructivist approaches of CLE and CSP with the use of CMC while keeping each participant responsible for his/her work can be addressed by the implementation of studio-based learning online.

The Differences Between Studio-Based Learning Online and Teamwork

The distinction has been made between SBL and traditional teamwork. Traditional teamwork usually refers to student collaboration in order to produce deliverables, while in SBL each participant remains responsible for his/her own project.

The characteristics of SBL are an iterative process that begins when participants build artifacts independently. Participants get together to present and critique these artifacts. As an iterative process, SBL allows for the application of feedback for the refinement of artifacts.

The Essentials of SBL

Brocato (2009, p. 139) defines SBL as: “an inquiry apprenticeship model that follows problem-based learning but allows a more pervasive person-centered approach.” Studio-Based Learning (SBL) is learner-centered: “SBL centers on projects instead of taking students through a series of tightly controlled problems” (Khun, 1998, p. 66). In SBL, “Students are actively involved in their learning process, interacting with others and forming connections with prior or consecutive knowledge” (Lynch & Markham, 2003, p.

71). In addition, "SBL teaching aims to foster creativity, reflection, articulation and reasoning, all of which are important life-long learning skills and valuable graduate attributes" (Simpson, Burmeister, Boykiw, & Zhu, 2003, p. 2).

Examples of the Studio-Based Learning (SBL) teaching method may be found in schools of architecture, where students undertake a design project under the supervision of their instructor. In this pedagogy, the instructor is not the focus of the class; students instead, work together to solve problems and answer each other's questions (Docherty, Sutton, Brereton, Kaplan, & Brown, 2000).

Brocato's (2009) research focused on watching the use of SBL in the school of architecture at the university of Colorado and she identified important features that made possible to move SBL into other disciplines. She identified the required process as "propose-critique-iterate," where proposing refers to the student's individual design, which is the central activity.

The SBL model for teaching and learning in IT traces back to Vygotsky's social constructivism, hosted in the school of architecture and followed in 2000 by the research of Jamieson, Docherty, and Brown (Lynch & Markham, 2003 p. 70), which brought SBL to be implemented in information technology instruction.

Goals of SBL Implementations in the Computer Science Classroom

SBL has been successfully implemented in computer science education for diverse purposes. For example, Estey, Long, Gooch, and Gooch, 2010; Hendrix, Myneni, Narayanan, and Ross, 2010; and Lynch, Carbone, Arnott, and Jamieson, 2002, were motivated by the industry requirement for soft skills in computer science graduates. They believed that the interaction afforded by SBL could better prepare students for jobs in the industry.

On the other hand are the educators (Stone & Madigan, 2008; Turner, Perez-Quinones, Edwards, & Chase, 2010), who aim to break the monotony of computer science courses with the introduction of SBL activities. They believed that SBL would help them increase student motivation.

There are others (Hamer et al., 2008; Hundhausen, Agrawal, Fairbrother, & Trevisan, 2009), who try to promote critical thinking skills among students, while helping them to become familiar with the learning process found in industry and research.

For the particulars of the work reported in this dissertation, the motivation is to provide better quality online education and to proactively help students overcome impasses caused by common CS1 errors. Based in these premises, it is important to revise common characteristics of CS1 courses as well as to identify widespread goals and challenges.

One widespread limitation of CS1 courses is that students rarely see code from their peers. Jonassen et al. (1995) and Reimer, Cennamo, and Douglas (2012) found that SBL interaction increases student self-confidence. Student-student interaction provides computer science education the possibility of innovative educational alternatives. One of such alternatives is the mastering of soft skills, which are in high demand for computer science professionals in the industry (Estey et al., 2010; Hundhausen, Narayanan, & Crosby, 2008).

New Perspectives on an Old Curriculum

The introductory sequence for the computer science curriculum has remained constant for several decades. The majority of institutions offer a two-course sequence, CS1 and CS2, known in several of the University of Hawai‘i campuses as ICS111 and ICS211. This sequence even while covering the same basic material, has evolved due to constant changes in technology and multiple pedagogical approaches.

The CS1 course is the cornerstone course for almost every degree in computer science. CS1 is the course where computer science fundamentals, such as programming and algorithms are introduced. Algorithms and programming courses present an obstacle to a significant number of potential computer science students and therefore, it has become the object of study by many researchers and faculty members alike (Ambrosio & Martins, 2011; Han & Beheshti, 2010; Stone & Madigan, 2008; Teague, 2009). Difficulties in learning programming often result in lack of motivation and high dropout and failure rates. This affects not just introductory programming courses but also the

computer science degree altogether (Hundhausen et al., 2008; Simon, Kinnunen, Porter, & Zazkis, 2010). Finding the precise pedagogy and correct tools to teach programming has been a common quest for computer science teachers for many years (Evans, 1996; Ficocelli & Gregg, 2011; Gross & Powers, 2005).

Algorithms and programming courses represent an obstacle for first year CS students (Ambrosio & Martins, 2011). Issues with programming include the difficulty of understanding the purpose of programs as well as difficulty grasping the syntax and semantics of a particular programming language (Teague, 2009). First year Computer Science students often encounter difficulties when learning to write procedural and object-oriented programs for the first time. This is also true of students being exposed to a new programming paradigm (Pillay, 2003).

High dropout, high failure rates and lack of motivation plague the first year CS scene. These are some of the reasons why CS1 has become the object of study of many researchers and faculty members. A plethora of pedagogies have been suggested to promote success in this class. To the same avail, computer science educators have developed software tools intended to ease this difficulty, better engage students and increase their likelihood of success in learning introductory programming.

With the goal of preventing dropouts, Whittington and Bills (2004) instituted a freshman seminar course to begin addressing student misperceptions about the importance of computer programming and to highlight the importance of introductory programming in the curriculum. They also created a special section of CS1 featuring a slower pace of lecture delivery.

Others have focused their efforts on student success. For example, Stone and Madigan (2008) focused on teaching CS for non-majors. They gave students personalized assignments related to their major. Students were given a vast array of project choices. Personalized assignments made students feel good about what they were learning, consequently increasing student success.

The above-mentioned approaches have changed the delivery pace and adapted the curriculum to student needs. SBL, however, offers a different perspective in the computer science scene. SBL has not been used to replace lectures or introduce new material, instead SBL has been used to reinforce lecture material (Carter & Hundhausen, 2011).

SBL online encourages online student collaboration, presentation of artifacts and critiques followed by the incorporation of feedback into individual work.

Outcomes of SBL Implementations in Computer Science Education

SBL has been implemented successfully in different courses. Considering only the SBL project, which is an NSF funded project focused on SBL implementation in computer science, SBL has been implemented in 25 courses in 15 institutions across the US. Comparisons from SBL to traditional instruction have shown increased retention in longitudinal studies done by the SBL project (www.studiobasedlearning.org). The SBL Project tracks retention by looking at the number of students who take all courses in a sequence, and the number of students who subsequently remain in the computing program, at each of the participating institutions.

Other researchers have implemented SBL without being part of the SBL project. These are their perspectives in regards to the pedagogy; Reimer et al. (2012), as well as Estey et al. (2010) observed a positive impact in student confidence and peer learning. On the other hand Docherty et al. (2000) and Simpson et al. (2003) claim that SBL provides students with industry-like experiences, by solving similar problems as those faced by professionals in their field.

SBL has been presented as a promising alternative for computer science. However, the research shows that little has been done for fully implementing SBL online within online courses. Research that has touched on this issue focuses mainly on providing a suitable online environment for peer reviews. These online environments act like an external activity for the reinforcement of course material.

Expertiza, RRAS and OSBLE are all examples of environments that support peer reviews. Expertiza consists of two parts, one of them, PeerGrader allows peer reviews while the other part saves student work in repositories for later use by other students (Gerhinger, Ehresman, Conger, & Wagle, 2008). RRAS was designed with the purpose of providing anonymous peer evaluation of assignments. RRAS can be used to manage peer evaluation of assignments with the guidance of a rubric designed and maintained by the instructor (Trivedi, Kar, & Patterson-McNeill, 2003). OSBLE (Online Studio Based

Learning Environment) was designed to support pedagogical code reviews (PCR). Students upload programming solutions into OSBLE to be reviewed online by peers. In contrast with PeerGrader, OSBLE provides the identities of both the author and the reviewer as dictated by industry standards (Hundhausen et al., 2009).

OSBLE is a system tailored to support pedagogical code reviews, which are an integral part of the implementation of Studio-based learning online as described by this dissertation. OSBLE centralizes activities related to code reviews by providing access to the code to be reviewed before the SBL session, as well as to students' reviews after the session. Hamer et al. (2011) affirm that OSBLE is 37% more efficient than paper, face-to-face code reviews. OSBLE has been in use by students taking face-to-face programming classes. On the other hand, SBL-Online was incorporated into an online programming class. SBL provides students with benefits which include fostering student creativity, reflection and reasoning (Docherty et al., 2000), improved student self-confidence (Estey et al., 2010), and real-world experiences (Jonassen et al., 1995; Simpson et al., 2003).

How SBL-Online Differs from Other SBL Implementations

Despite the multiple SBL implementations, none of them thus far have described how SBL sessions are designed, how material for these sessions is selected or how SBL sessions are placed into the course timeline. SBL sessions seems were simply introduced at convenient times featuring assignments that display an array of possible solutions, but no other restrictions. Based on these premises, the differences between previous work and this approach are as follows:

The courses selected for this study were delivered entirely online. At the time of this writing, SBL-Online is the only fully online SBL implementation. Online code critiques have been done (Hundhausen, Agrawal, & Ryan, 2010) but they took place as a side activity in a traditional face-to-face course.

Identifying challenging material is the first step. Assignments to be critiqued during SBL sessions have the special characteristic that they must be solvable following a variety of thinking paths. For this study, strategic building of assignments is required. In

addition to having multiple potential solutions, assignments include material that was challenging for previous groups of CS1 online students.

This study requires SBL-Online sessions to be placed at key times during the semester. To maximize the impact, SBL-Online sessions were strategically scheduled after the most confusing and error-provoking concepts were introduced.

Recording of student interaction during SBL-Online sessions provides an insight into the inner workings of an SBL-Online session.

SBL-Online, as presented in this dissertation, consists of three fundamental elements; the online nature of the course's implementation environment, the identification of common errors made by students and the studio-based learning (SBL) pedagogy. For this approach, common Computer Science 1 (CS1) student errors have been identified. Points in the curriculum where students consistently tend to make the most mistakes were labeled as key. Identical previous courses have been revisited to identify concentrations of errors throughout the semester. SBL addresses these errors with interventions strategically placed during the semester timeline. In short, SBL-Online is the implementation of SBL techniques at key times in an online course.

Summary

SBL pedagogies have been successfully implemented in computer science education. SBL pedagogies are based on social constructivism providing an alternative to traditional education. SBL-Online expands the benefits of SBL into online education. Furthermore, SBL-Online was designed to address recurring errors that students tend to make in CS1 with the goal to proactively engage students in peer evaluations designed to promote learning.

Chapter 3. SBL-Online Revisited

This chapter begins with the SBL theoretical background and components. It continues with relevant theories regarding student learning in online environments. It addresses the evolution of CS1 as well as its known problems, and to finish, it addresses common student errors and how they are used to create SBL-Online assignments.

SBL-Online consists of three fundamental elements (Figure 1); the online nature of the course, the identification of error clusters during the student's semester-long learning path and the studio-based learning pedagogy.

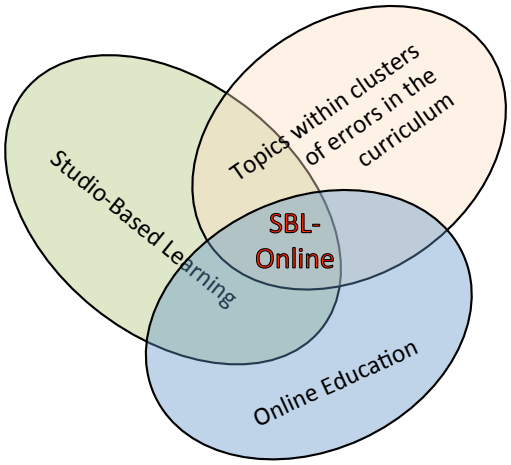


Figure 1. SBL-Online

This literature review will cover these fundamental elements as well as closely-related topics as needed in order to answer the research questions, which are given in the following section.

The Purpose of This Study

The purpose of this study is to describe how SBL-Online impacts student's behavior, attitude towards the course and semester grades in an online CS1 course. This study is addressed in parts, as described by the following research questions:

Is there any indication of how the application of SBL-Online aids a beginning programming class?

During SBL-Online sessions,

- What are the interaction highlights?
- Can indicators of learning be documented?
- Are students' misconceptions being identified?
- Are students' misconceptions addressed?
- Are students acquiring knowledge that will transfer to other course activities?

How SBL Became Part of the Computer Science Curriculum

Although the traditional model of learning is still widely used in computer science, constructivist pedagogies have made their way into the computer science classroom. Social constructivism recognizes the importance of social interaction in the learning process. Social constructivism is about the exchange of opinions and socially negotiating the meaning of observable facts. Construction of knowledge is the result of an active process of articulation, reflection and collaboration among learners, and "collaboration aids in developing, testing, and evaluating different beliefs and hypotheses within learning contexts. Through the process of articulating covert processes and strategies, learners are able to build new and modify existing knowledge structures" (Jonassen, Davidson, Collins, Campbell, & Haag, 1995). In social constructivism knowledge becomes commonplace: "Learning is conversation, and the thinking and intelligence of a community of performers or learners is distributed throughout the group" (Jonassen et al., 1995).

One example of social constructivism is peer reviews, which has been perceived as beneficial to computer science students. The term "peer review" has been defined as the process of students looking at other students' work, and providing comments (Hamer, Luxton-Reilly, Purchase, & Sheard, 2011). There have been many studies depicting the benefits of peer reviews. Hundhausen et al. (2009) argue that peer reviews are potentially valuable and important for students because they represent an authentic, real-world experience. Peer reviews promote soft skills, such as: presenting code to their peers and learning to deal with criticism. Students also learn to provide constructive criticism to their peers. Peer reviews promote opportunities for deep learning and critical thinking. Peer reviews have the potential to promote student's learning and intellectual development to higher levels of the Bloom's taxonomy. Turner, Pérez-Quiñones, Edwards, and Chase (2010) point out that according to Piaget and Vygotsky peer reviews are active processes that provide opportunities for dialog among students and promote learning while providing new perspectives to challenge current believes.

According to Wolz et al. (1997) students like to look good in front of their peers, therefore peer reviews become an incentive for students to succeed. Reily, Finnerty, and

Terveen (2009) explain that participating in peer reviews often result in better learning outcomes.

Peer reviews are an integral component of the SBL pedagogy. Lackney places the origin of SBL in America back to John Dewey's Laboratory School in Chicago in the late 1800s (as quoted by Bray, 2012). SBL has been used mostly in art disciplines, which are wicked by nature. Wicked disciplines present problems that do not have only one correct solution. Computer science is a wicked discipline therefore it allowed for the implementation of SBL techniques (Docherty, Sutton, Brereton, & Kaplan, 2001). According to Lynch et al. (2002), prior to 2002 Studio-Based Learning was not used much in computer science. In later years, however, there has been strong and growing support for the implementation of courses that contain more hands-on learning where students will learn as they solve problems (Gonsalvez & Atchison, 2000).

Computer science education researchers highlight the varied perceived benefits of SBL. Gonsalvez and Atchison (2000) draw attention to the hands-on component of SBL which provide students with experiential learning. Hamer et al. (2008) emphasize the benefits of SBL as a pedagogy where students teach and learn from others. It has been claimed that students acquire reflective, critical and creative skills as they collaborate during SBL sessions (Hamer et al., 2011).

Many claims have been made about the benefits of SBL in computer science education. Therefore is important to explore SBL characteristics to identify components and implementations, as well as to identify the good and the bad traits in SBL.

SBL Components in Programming Courses

The following is a list of participant roles within an SBL implementation focused on software inspection. These are the roles that most resemble those played by computer science students as they review programming assignments during studio sessions. This is a general list as presented by Gilb and Graham (1993):

- Author—the person whose program is being inspected; This person must explain the code at the beginning of the session and may be asked to clarify points as needed. This is the main role within SBL-Online.

• Reader—reads the Author's code solution aloud. This role is seldom used in computer science education and it is not implemented by SBL-Online.
• Inspector—inspects the code guided by the provided code inspection checklist. In SBL-Online more than one student can be inspectors.
• Recorder—documents the code inspection. In SBL-Online, this person also submits the code inspection checklist at the end of the session.

The SBL Design Crit

Code inspectors are in some instances called design crits. Participants other than the author may play the role of design crit. Kehoe (as cited by (Hundhausen, Fairbrother, & Petre, 2011) explains that during software inspection, the design crit is a role shared by the reader and the inspector. He emphasizes that an SBL design crit is different in nature from other forms of learning discussions because the evaluation takes place in public. Pat Donahue, member of the SBL project, describes students as apprentices working under the watchful eye of a master (instructor). In this setting, students interact as needed providing periodic critiques of student projects while working to perfect their product (Bray, 2012). Critiques of this kind aim both to influence the trajectory of the work and to provide opportunities for students to learn from each other.

As part of the SBL process, critiques in programming courses involve doing individual work before participating in the studio session. Students must inspect the author's code in advance. This process helps students identify problems for later discussion with their peers. Lister (2001) notes that performing an individual code inspection helps students prepare and feel confident, he believes that upon completion of their asynchronous activities, students feel more prepared and knowledgeable when they come to class.

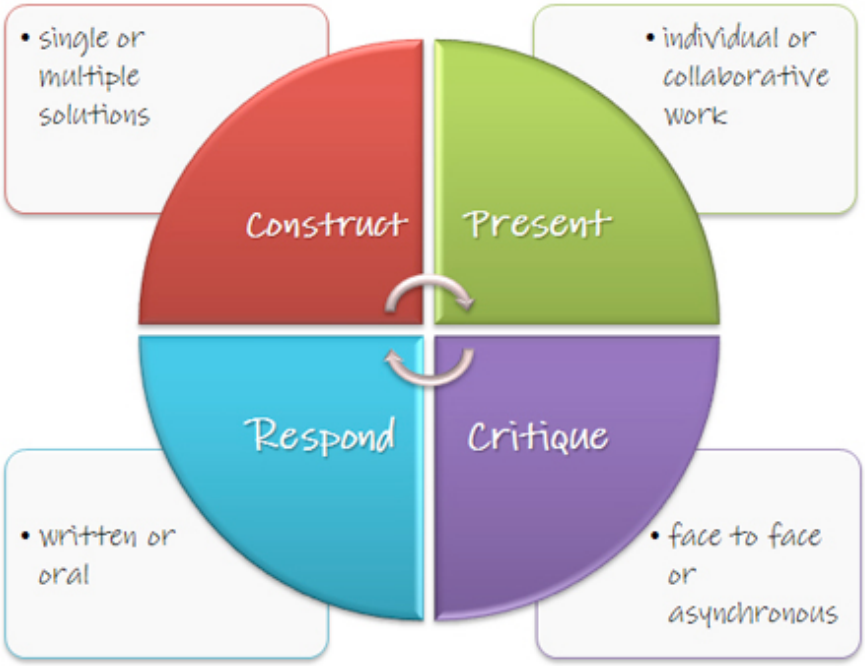


Figure 2. The SBL Process

As shown in Figure 2, many different combinations of SBL components are possible. However, as long as there is a construct, a presentation, a response and a critique, it can be considered SBL.

An important characteristic of SBL to keep in mind is that assignments to be critiqued must be complex, meaningful projects that can be implemented in different ways, with a wide range of valid solutions (Khun, 1998). This gives students the opportunity to be innovative and creative.

Effects of SBL on Students

Being able to provide peers with software critiques produce varied effects on students. Students who experience SBL might see other students and themselves as legitimate sources of knowledge. During this process they might recognize the multiplicity of correct solutions to problems and the degrees of quality of these solutions, instead of seeing every problem as having a right or wrong answer with only one correct solution. With practice, students become more comfortable with SBL and it becomes easier for them to offer their own critiques and suggestions. Hundhausen, Agarwal,

Trevisan (2011) emphasize that "one gains competence within a community of practice by having opportunities to participate, in increasingly central ways, in the practices of the community" (p. 118). Hundhausen and Brown (2005) noticed that SBL stimulated their students to share and discuss their knowledge while increasing student motivation and level of interest in the course material.

When students collaborate, share and discuss ideas they acquire soft skills, which were overlooked in the past. However, more recently, jobs in the computing field increasingly require design, critical thinking, communication, and the ability to collaborate effectively with coworkers in addition to technical skills (Hundhausen, Agrawal, & Ryan, 2010; Lynch et al., 2002).

Computer science educators have explored the use of studio-based learning in individual computing courses (Carter & Hundhausen, 2011; Hundhausen, Agrawal, Fairbrother, & Trevisan, 2010; Reimer, Cennamo, & Douglas, 2012) while others have implemented it throughout entire degree programs (Docherty et al., 2001; Docherty, Sutton, Brereton, Kaplan, & Brown, 2000). SBL has not been used to replace lectures or introduce new material, instead SBL has been used to reinforce lecture material (Carter & Hundhausen, 2011).

> In addition to requiring students to actively engage in the learning process, this activity (SBL) enables students to exercise a variety of skills that are increasingly important for computing careers, including
> (a) computer programming,
> (b) critical review of their own and others' code, and
> (c) communication and teamwork
> … by having students review each other's code in teams, pedagogical code reviews strive to promote a sense of community, and to build students' communication and teamwork skills, which are often neglected in traditional lower-division computer science courses. (Hundhausen et al., 2009, p. 291)

SBL Implementations

The following is a list of some SBL implementations within computer science or information technology curricula. These implementations are listed in two categories: the first one lists full degrees that implemented the SBL pedagogy throughout the curriculum; the second is composed of independent course implementations.

SBL-Based Degrees

Two extensive implementations of SBL throughout whole degree programs are the Bachelor of Information Environments (BIE) at the University of Queensland (Docherty et al., 2001) and the Bachelor of Information Management and Systems (BIMS) degree at Monash University (Carbone & Sheard, 2002). Both degrees were created based in the architectural studio model. The BIE degree combines computer science and design, while the BIMS degree focuses in computer science education. In both degrees, SBL is implemented throughout the curriculum.

SBL Courses

There are 25 SBL courses within the SBL project. The SBL project encompasses fifteen institutions across seven states in the US. The scope of the SBL project as of October of 2012 is illustrated in Figure 3.

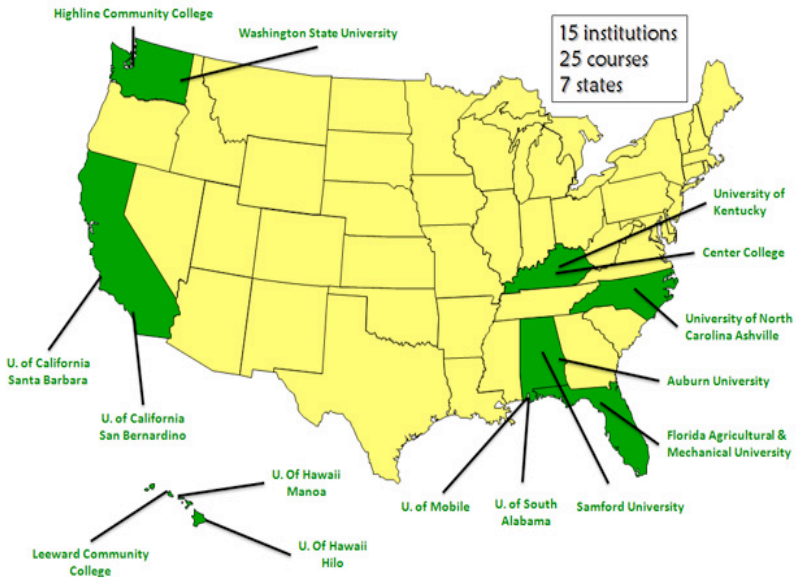


Figure 3. The Scope of the SBL Project (studiobasedlearning.org)

"Introduction to Computer Science I" at Leeward Community College is part of this study. There are, however, other SBL implementations, which are not part of the SBL project. These are some examples:

- Gannod, Burge, and Helmick (2008) used SBL in their implementation of the inverted classroom pedagogy to teach a software engineering course.
- Ungar and White (2008) used SBL and agile user-centered design within Human-Computer interaction courses.
- Gonsalvez and Atchison (2000) implemented SBL for experiential learning, in two IT-based undergraduate courses.
- Ludi, Natarajan, and Reichlmayr (2005) used SBL via wireless notebooks as part of a Java programming course.
- Reimer et al. (2012) cite two more SBL implementations. In one of them Nurkkala and Brandle integrated the SBL model over the span of multiple semesters to address long-standing gaps in the typical software engineering education.

SBL Essentials

With a plethora of multiple SBL implementations it is necessary to emphasize that even within this diversity, all SBL implementations continue to be similar in their foundations because they share common characteristics. Hendrix, Myneni, Narayanan, and Ross (2010) explain the SBL process as follows:

> (a) Students are given non-trivial problems for which they have to design and implement computational solutions individually or in groups.
> (b) These problems are amenable to multiple solution strategies. Thus, students have to consider alternate solutions and their tradeoffs in terms of efficiency and software engineering considerations, choose the best, and justify their choice.
> (c) Students must articulate their solutions and justifications to other students for peer review, feedback, and discussion. This can be done orally, in writing, or both.
> (d) Their peers and the course instructor must provide comments and criticisms, again in writing, orally, or both.
> (e) Students are given the opportunity to respond to this feedback and modify their solutions appropriately. (Hendrix et al., 2010, p. 505)

As expected, given the common characteristics shared by SBL, very similar processes are given by Hundhausen, Agrawal, and Ryan (2010) and by Pat Donahue (Bray, 2012).

The Benefits of SBL

A variety of beneficial SBL attributes as perceived by different researchers in the field. For example, Docherty et al. (2001) see SBL as a tool to foster lifelong learning

skills such as creativity, reflection, articulation and reasoning. They argue that SBL also affords students the opportunity to solve problems similar to those solved by professionals in their field. These beliefs are shared by Simpson, Burmeister, Boykiw, and Zhu (2003), who believe that a slow and positive exposure of students to real-world projects, which prepare them for real life. Others like Reimer et al. (2012) and Estey et al. (2010) found that SBL has a positive impact in self-confidence and peer-learning. While Jonassen et al. (1995) believe that the best experiences for students are those that provide them with the opportunity to work and interact with their peers while mastering new knowledge.

Motivations for Implementing SBL

Given the perceived benefits of SBL pedagogies it is understandable that SBL found a place in computer science education. Educators that implemented SBL had different motivations for doing so, based on different expected outcomes. The following is a list of the motivations researchers had for implementing SBL techniques in their classrooms.

- The motivation for Estey et al. (2010) was to provide students with industry-like experience.
- The motivation for Hendrix et al. (2010) and Hundhausen, Agrawal, and Ryan (2010) was to provide students with the soft skills required by employers.
- The motivation for Lynch et al. (2002) was the desire to better prepare students for jobs in industry.
- The motivation for Turner et al. (2010) was to break the monotony of a class in a typical lecture format.
- The motivation for Stone and Madigan (2008) was to engage and motivate students.
- The motivation for Hamer et al. (2008) was for students to become familiar with the learning process found in industry and research.
- The motivation for Hundhausen, Agrawal, Fairbrother, and Trevisan (2010) was to promote critical thinking skills among students.

In the same vein, the motivation for this study is to provide better quality online education and to proactively help students overcome impasses caused by common CS1 errors.

Problems Derived from SBL

As motivated educators implemented SBL attempting to capitalize on potential benefits, they realized that there are also a variety of problems that may arise within or as a result of SBL implementations. An example of this arises when students do not inspect assigned code individually before meeting for the SBL session. This potentially diminishes SBL's beneficial outcomes (Hundhausen, Agrawal, & Ryan, 2010).

Even when reviews are done in advance, Wang found out that reviewers' lack of knowledge might negatively impact SBL peer reviews. On the other hand, poorly written programs usually are difficult and boring for skilled reviewers to carefully inspect. Some students lack the programming ability to be qualified reviewers so they either give few or confusing comments (Wang, Yijun, Collins, & Liu, 2008), while others experience difficulties rating and classifying their peers solutions (Denning et al., 2007).

Overall, research shows that students were intimidated and uncomfortable performing peer reviews (Estey et al., 2010). Denning et al. (2007) found that peer reflection caused apprehension in 47% of their students whom felt intimidated knowing that their code would be critiqued. Choy and Ng found out that students were more willing to engage in question-and-answer discussion than to share artifacts (as cited by Hamer et al., 2011).

Aside from the positive and negative byproducts of SBL implementation previously articulated, there are other factors that may lead to an untrustworthy review (Reily et al., 2009):

1. Retaliation. One student may give a negative review to another in retaliation for a perceived undeserved negative review.
2. Collusion. Students agree in advance to give each other a good review regardless of the code's quality.
3. Competition. A student may give harsh negative reviews in an attempt to skew the grading curve.
4. Laziness. It takes effort to perform a thorough review.

So far, we have explored the SBL pedagogy and we have seen the motivating factors that resulted in a wide implementation of SBL in computer science education. We have

also reviewed the advantages and pitfalls of SBL implementation. Now, we will focus on online education as a platform for SBL-Online implementation.

Online Learning Overview

This online learning overview is a brief general description of the relevant theories behind distance learning and the use of technology in education as they relate to this study.

Distance Education

Moore (1990) defined distance education in a very generic way as instruction delivered to persons located in different time and place from the instructor. Today, in a more generic term, online education can be defined as "educational computer networking independent of pedagogical design, educational level or field of study. Online education covers everything from formal course activities to informal peer networking and interaction" (Harasim, 1996, p. 206).

Educational institutions are trying to provide better quality online education arousing research interest in effectiveness of multimedia. Research done by Abdelhakim and Shirmohammadi (2007) reveals that student-student and student-teacher interaction can increase with the use of videoconferences as part of distance education courses. Videoconferencing software usually provides means for video, audio and text-based communications all at once (Heeler & Hardy, 2005). Carrington, Kim, and Strooper (2010) found that by enhancing the quality and convenience of communication, student satisfaction improves increasing student success and promoting positive learning outcomes.

Communication between teacher and student is also a vital element of successful distance education. Moore (1990) rates the success of distance education on the quality of the content of the dialog between teacher and student and the effectiveness of the communication system in an educational process. Lister (1998) argues that interaction enhances distance education, promotes earlier completion of coursework, enhances performance on tests and increases student retention rates.

The Influence of Technology

The success of online interaction cannot escape its dependence on the underlying technology that supports it. There has been an extensive debate over how the use of technology by teachers and students may affect student learning (Kanuka, 2008). On one hand, Jonassen (1996) argues that educators reshape education, while technologies do not. On the other hand, McLuhan and Fiore (1962) wrote the book, "The Medium is the Message," highlighting technology as an intrinsic component which shapes the message. Middle ground is found with Chandler (1996), who claims that when we interact with media, we shape and we influence our message but we also are shaped and influenced by the media. The part of this study that addresses student interaction during SBL-Online session falls into Chandler's paradigm.

SBL became part of computer science classrooms along the US. As stated before, SBL has been applied in different courses, however, this study focuses in the CS1 course, and therefore, it is relevant to revisit it in more detail.

The Evolution of CS1

Over thirty years ago the ACM Computing Curricula designated the terms "CS1" and "CS2" for the first two introductory courses of a computer science major. Since then, introduction to programming courses are referred to as CS1 and basic data structures courses as CS2. This common shorthand has been widely used to enable students to transfer between institutions and as a base of many research studies (Hertz, 2010).

With the popularity of computers, today, computer programming is seen as "the new Latin" of the school syllabus. In 2006 Java became the most used programming language in CS1 (Schulte & Bennedsen, 2006), and it is taught in many universities across the US (Ficocelli & Gregg, 2011). Students in varied majors are required to take this course. In this course students are expected to develop programming skills, usually through a series of individual programming assignments (Hundhausen et al., 2008). CS1 is expected to help students develop general problem-solving skills through learning programming (Sleeman, 1986).

Debates About CS1 Pedagogies

Throughout the history of CS1 there have been many debates about computer science pedagogies (Ficocelli & Gregg, 2011). These debates usually concern many issues such as the choice of programming language, the order in which the topics are covered, and even if the use of object oriented programming as a viable technique for teaching programming concepts. Schaub (2009) lists different ways to teach computer science effectively, including variations in pedagogies that include GUI development, web development, use of robotics, microworlds and even returning to the use of console applications.

To the same avail, researchers have tried to identify disagreements among CS faculty. Evans (1996) performed a survey in which he interviewed computer science (CS) faculty in Germany, Denmark and the US. The interview required teachers to specify whether object-oriented-concepts were covered or if they only taught imperative programming. This same study inquired about the order in which topics should be introduced. Results from this study show that most CS professors have different opinions about how a CS1 course should be conducted as well as the order in which topics should be introduced. Hertz, (2010) surveyed CS1 and CS2 instructors revealing the disagreement on which topics are the most important for students to master during these courses. These disagreements may be a byproduct of the high attrition experienced by CS programs around the US.

Reasons for High Attrition in CS Programs

At the beginning of most CS degrees, students are introduced to computer programming usually in the CS1-CS2 sequence as mentioned before. Research shows that first year Computer Science students often encounter difficulties when learning to write procedural and object-oriented programs for the first time. This is also true of students with previous programming experience being exposed to a new programming paradigm (Pillay, 2003).

Algorithms and programming courses represent an obstacle for first year CS students (Ambrosio & Martins, 2011). Issues with programming include the difficulty of

understanding the purpose of programs as well as difficulty grasping the syntax and semantics of a particular programming language (Teague, 2009).

Many studies have shown that learning to program is difficult for many students (Hanks & Brandt, 2009). Guzdial, Hohmann, Konneman, Walton, and Soloway (1998) argue that programming is hard to learn because much of the thinking behind a program is implicit and the process of programming is long and complicated. Difficult material plays a role in dropout rates for any discipline. In the case of computer science, programming is a difficult skill to learn (Gross & Powers, 2005).

High attrition however has also been attributed to other facts. For example, Whittington and Bills (2004) argue that dropouts may happen because most of the students are not interested in careers in computer programming but in other aspects of computing. To address this issue, they instituted a freshman seminar course to begin addressing student misperceptions about the importance of computer programming and to highlight the importance of introductory programming in the curriculum. They believe that if students stop perceiving CS1 as a hurdle and more as an important foundation, student success will increase.

Others have focused their efforts on retention of non-CS majors required to take CS1, such as electrical engineers (Stone & Madigan, 2008). Stone and Madigan gave students personalized assignments related to their major, which made students feel good about what they were learning. On the other hand Whittington and Bills (2004) focused their efforts on slowing down the pace of CS1 for students that needed extra time to assimilate course content. They made two sections, one of them at a slow pace, while a regular section of CS1 progressed at regular speed.

There are those who believe that student background plays a role in student success. In this regard, Han and Beheshti (2010) attribute the high dropout rates on the lack of math and logical reasoning skills. Rountree, Rountree, Robins, and Hannah (2004), on the other hand, have tried with limited success to predict performance based on student status, educational background and student expectations.

Others like Barland (2008) argue that reordering course material can help student success. He argues that the problem may be because Java topics are introduced without giving a strong rationale for why they are needed leaving students wondering. Situations

like this may prompt students to believe that "programming is not about problem-solving, but rather about putting certain keywords in the right place" (p. 256).

Perspectives on Programming Errors

Programming errors have been the object of research for educators and cognitive scientists (Spohrer & Soloway 1986). Educators want to improve students' understanding of programming. Cognitive scientists want to learn how programmers use their problem-solving skills when facing complex problems as well as how misconceptions and cognitive constraints may lead to errors.

Programming errors may be the result of students lack of problem solving skills. So, even students may be familiar with programming syntax, facts and procedures they have difficulty thinking critically as well as analyzing and solving real-world problems. Guzdial et al. (1998), believe that errors in CS1 are the product of the difficulty that students have on decomposing the problem to better understand it and on integrating a solution in a viable way.

On the other hand, some students may do well in exams, however, they have trouble applying acquired knowledge into new situations (Simpson et al., 2003). This may be because students have failed to connect new programming problems to past programming experiences or perhaps failed to identify abstract reusable patterns (Linn, 1996).

Others like Edirisinghe (2008) believe that the problem lays in the programming process. He argues that the reason why introductory computer programming remains a challenge is because students have difficulty identifying and correcting errors in their own code even with the help of debuggers. This is attributed to the lack of understanding about the cause of their own errors. Edwards (2003) found out that students learning to program may struggle with comprehension and analysis skills. They are under the impression that once a program runs it should be correct. However, this is seldom the case. Students believe that most programming errors are reported by the compiler, therefore, instead of using critical thinking and exhaustive code testing, they randomly modify the code until it works. Once the code is fixed, usually students aren't sure why it works or why it wasn't working before. As time goes by, students continue to make the same mistakes over and over.

Others blame errors on student misconceptions about language constructs, but Spohrer and Soloway (1986) argue that misconceptions about language constructs are not as widespread as typically believed.

Recurring errors is reportedly a widespread problem plaguing the CS1 scene. This is the reason why many pedagogies have been attempted in this course. SBL can be seen as an appropriate solution given that teaching programming skills is often compared to teaching writing. It can better be accomplished by a repeated cycle of writing, getting informed feedback and rewriting (Woodley & Kamin, 2007).

Giving feedback to students may seem to be the appropriate action to follow. However, it also has a downside. For example, Linn (1996) found that when active learners are exposed to correct solutions they learn to use these solutions, but cannot distinguish correct and incorrect solutions in a multiple choice setting. In other cases, when the program works, students tend not to pay attention to how it was designed (Turner et al., 2010). These examples show us that giving feedback, while useful, may not be the perfect solution in every case.

Feedback is just one approach aimed to correct and eliminate errors that students make. Errors have lingered within CS1 stubbornly and it is this what motivates researchers to try different approaches to rid of them.

CS1 Error Categorizations

The frequency and multiplicity of errors within CS1 has motivated computer science education researchers to categorize and quantify student errors as well as to pinpoint difficult areas in the CS1 curriculum. For example Barr, Holden, Phillips, and Greening (1999) sent out a survey for computer science faculty to find out which topics were those that faculty perceive being the hardest for students.

In 2003 Hristova, Misra, Rutter, and Mercuri (2003) conducted a survey of college professors, teaching assistants and students in computer science. They asked them to identify the five most commonly made programming errors and the hardest errors to find and fix. This yielded 62 different errors which they filtered selecting the twenty errors that they considered essential to the educational aspect of their project.

On the other hand, a Delphi study was implemented by Kaczmarczyk, Petrick, East,

and Herman (2010) with the participation of programming instructors to identify student's troublesome issues.

While some categorizations are too specific others are too wide. Attempts have been made to categorize student errors in order to better target them, however the simplest of categorizations and the one used by Hristova et al. (2003) is the following:

Syntax Errors: They usually cause the code not to compile, making them easily identifiable. They are spelling mistakes, punctuation and instruction order.

Semantic Errors: They occur on a more abstract level than syntax errors. They deal with the meaning of the code and derive from mistaken ideas of how the language interprets certain instructions.

Logic Errors: They are most often the result of erroneous thinking by the programmer rather than referring to the programming language.

Different error categorizations were revisited in search for a categorization suitable for this study. While some categories were too broad, others were too detailed. It was necessary to find middle ground. Robins, Haden, and Garner (2006) found out that the main tradeoff involved in designing the list of errors was between richness and practicality. A rich list implied a detailed classification of many problem types, while a practical list suggests an easy to master, compact list of problem codes. Robins et al. have put into operation a well-categorized and complete list of errors, which they have been using since 2003. They began compiling this list years before (Garner, Haden, & Robins, 2005). The granularity of the error list produced is ideal for this study.

In order to create this error categorization, Robins et al. (2006) collected data from questions that students asked the teaching-assistants (TAs) during their programming laboratory sessions in a CS1 course. When students asked the TA for help, the TA will help the student out, recording and categorizing the kind of issue that the student had.

As the semester progressed the kind of questions asked by the students evolved. Some errors were no longer an issue while others remained, and new ones emerged. This study helped them figure out the lifecycle of errors. They also found that some student misconceptions remained active until the end of the semester. Over all, they were able to graph the incidence of different types of errors throughout the course.

	Issues Causing Problems
B1	Tools, operating system issues, lost files, web-posting issues, file permissions.
B2	Understanding the task.
B3	Stuck on program design, students know what to do but they don't know how to do it
G1	Problems with basic structure, code outside methods, data fields outside the classes, etc.
G2	Problems with basic object concepts such as creating and using instance objects.
G3	Problem naming variables, constant or methods suggesting that they do not understand their purpose
G4	Trivial mechanics, braces, brackets, semi-colons, typos, spelling, naming conventions, import statements forgotten, output formatting, the use of magic numbers.
S1	Control flow, basic sequential flow of control. The role of the main and init methods. Call and return of methods.
S2	Loops, related to repetition, loop headers, bodies and blocks
S3	Selection, if-else and switch, blocks related to selection and validation.
S4	Booleans and conditions, formulating and understanding Boolean expressions (doesn't include precedence)
S5	Exceptions, throw/catch
S6	Method signatures and overloading
S7	Data flow and method header mechanics includes arguments, parameters and return-type values. Mismatching parameter specs, incorrect return values, incorrect use of void.
S8	Terminal or file I/O.
S9	Strings and string methods
S10	Arrays as data structures, subscripts, contents, declaration and initialization. It includes ArrayLists.
S11	Variable initialization and assignment

S12	Visibility and scope, reuse of identifiers
S13	Expressions and calculations, precedence and random number generation and use
S14	Data types and casting
S15	Reference types, the behavior of reference types being different from that of primitive data types.
S16	Class vs instance, the use of "static"
S17	Accessors and modifier methods their purpose and use
S18	Constructors concept and purpose
S19	Hierarchies, inheritance, extends, shadowing, super
S20	GUI mechanics, action performed and action listeners.
S21	Event-driven programming, flow of control in an applet.

Figure 4. Topics Causing Errors in CS1

The error categorization in Figure 4 was deemed ideal for this study. It displays an adequate granularity and is based on the use of the Java programming language, which is the one used in this study. Once the error categorization was adopted, error collection and categorization within this study gave way to the identification of clusters of recurrent errors within CS1.

Identifying Clusters of Recurrent Errors in CS1

The term vital topic in this dissertation refers to those topics in the curriculum that cause students to make frequent and recurrent-errors. Recurrent errors caused by vital topics are usually the ones that prompt students to reconsider if they want to continue pursuing a degree in computer science. Key points in the curriculum were identified by collecting and categorizing errors in CS1 and then, assigning them to the time where concepts related to those errors were first introduced. Topics causing the most errors were tagged as vital. The points in time when these errors occurred are labeled as "key points." It is in those points during the semester where SBL techniques are applied.

SBL-Online Defined

An SBL-Online session is defined as an SBL session taking place online at a key time during the semester. Lister (1998) identified a predecessor of SBL-Online in 1998. He proposed to emulate the studio at a distance to reap the benefits of interactive learning for those off-campus students. Lister (2001) described the next generation studio as virtual studio classrooms where students interact, and learn together with the use and support of multimedia technologies. Lister's description is very similar to the foundations for SBL-Online courses.

It was during this study that SBL techniques were implemented in an online CS1 course for the first time. However, as mentioned in chapter 2, pedagogical code reviews have already been done online by Hundhausen's OSBLE project (Hundhausen, Agrawal, & Ryan, 2010).

Preliminary Outcomes

Research outcomes from the SBL Project indicate that Studio-based learning has helped students to get to know each other better. SBL provides students with opportunities to share their experiences while writing code as well as to take a look at code developed by their peers and to give an opinion about it. This is especially enlightening when assignment solutions are radically different from one another.

After comparing a course using SBL with a traditional CS2 course, Hendrix et al. (2010) found that the SBL model greatly increased programming success as well as content mastery. They are optimistic that the SBL model will help inject new life to computer science education.

Summary

This chapter encompasses the literature review for this study. It identified and listed the various causes for students to dropout of CS programs. It continued with a brief discussion on different pedagogies with the common goal of preventing student dropouts from CS programs. An overview of the literature supporting the creation of SBL-Online and its implementation as a pedagogy geared towards breaking the cycle of recurring errors in CS1 was provided. The causes of these errors were identified and used to mark key points in the curriculum where SBL-Online sessions must take place to maximize effectiveness.

Chapter 4. Methodology

Introduction

The purpose of this chapter is to describe the methodology and instruments used with the goal of answering the research questions posed for this study. With this goal in mind, this chapter describes, the population and instruments used, as well as the setup for this study. We will begin this chapter by stating the purpose of this study, followed by a brief overview of the SBL Project as it relates to this study. The population and settings for this study will be described along with tools and instrumentation. This chapter ends with a description of preliminary outcomes of this study.

In order to measure the outcomes and effects of SBL-Online, some tools have been borrowed from the SBL project. The SBL project compares traditional vs studio-based learning (SBL) implementations of the same course and so does part of this study. The other part of this study consists of observing and analyzing events that transpire during SBL-Online sessions and their effects on students. For this purpose, video recordings as well as thick data collection are used.

SBL-Online Research Questions

The following are the research questions that guide this study. The purpose of this study is to describe how SBL Online impacts student's behavior, attitude towards the course and semester grades in an online CS1 course. This study is addressed in parts, as described by the following research questions:

Is there any indication of how the application of SBL-Online aids a beginning programming class?

During the SBL-Online sessions,

- What are the interaction highlights?
- Can indicators of learning be documented?
- Are students' misconceptions being identified?
- Are students' misconceptions addressed?

- Are students acquiring knowledge that will transfer to other course activities?

This project is heavily related to the SBL project. This chapter will provide the exact nature of this relationship as well as the extent of the influence that the SBL project posed over this research study.

SBL Project Overview

The SBL project (Hundhausen et al., 2008) project began as a multi-institutional effort comprising Washington State University (WSU), Auburn University (AU),and the University of Hawai‘i at Mānoa (UHM). The purpose of this project was to build a community of computing educators around the studio-based pedagogy to enhance computer science education. The original SBL project members planned to recruit computing educators to build a community of practice and to collaborate regularly via a web-based support system.

SBL Project Research Questions and Their Relationship to SBL-Online

There are five overarching research questions for the SBL project and how they relate to SBL-Online:

RQ1. Do students learn better in studio-based instruction than in traditional instruction?

In order to answer this research question The SBL project requires a pre- and post-test instrument, specific to each course. This study does not use this instrument.

RQ2. Are students able to better transfer and apply the knowledge learned through the studio-based approach in future courses?

The SBL project compares the performance in subsequent courses of students who take previous courses in the traditional and studio-based formats. This question is not addressed by this study. A variation of this question is posed. The SBL-Online question is: are students acquiring knowledge that will transfer to other course activities?

RQ3. Are students more engaged, invested and motivated in studio-based classrooms?

The SBL project tracks the attitudes, engagement, motivation and future plans of students in the consecutive courses at each university through both pre and post surveys using a modified version of the motivated strategies for learning questionnaire (MSLQ) and an end-of-semester questionnaire. This study makes use of these instruments as well, but to describe how SBL Online impacts student's behavior, attitude towards the course and semester grades in an online CS1 course.

RQ4. What are the long-term impacts of this approach in terms of persistence in the undergraduate program and future plans for computing careers?

SBL Project tracks retention by looking at the number of students who take all courses in a sequence, and the number of students who subsequently remain in the computing program, at each of the participating institutions. This question is not addressed by this study.

As stated before, modified versions of the above question RQ1, RQ2 and RQ3 are addressed within this study. Using the same methodology that the SBL project provides a more robust design and instrumentation, since their methodology has already been tested. Given that this study is part of the SBL project, required standard data collection had to take place. However, additional data was collected. The perspective taken to analyze this data is part of what makes this study unique.

Now that the relationship between the SBL project and this project has been established, it is necessary to focus on the unique characteristics of this study. The population addressed by this study and the tools used are described next.

Population and Sample Description

Leeward Community College is located in Pearl City, Hawai‘i. Leeward CC offers more than 60 degrees and certificates. Leeward CC also prepares students that wish to transfer to four-year universities. The associate in sciences degree in information and computer science is one of these degrees (http://www.leeward.hawaii.edu/about).

Students taking ICS111 at Leeward CC are in the most part Non-ICS majors, which is the trend in many universities (Pillay, 2003). Figure 5 shows the distribution between ICS majors and non-majors through the semesters where information for this study was

collected. The percentage of ICS Majors varies from 15.79% in the Spring2008 to 43.86% in the Fall 2010.

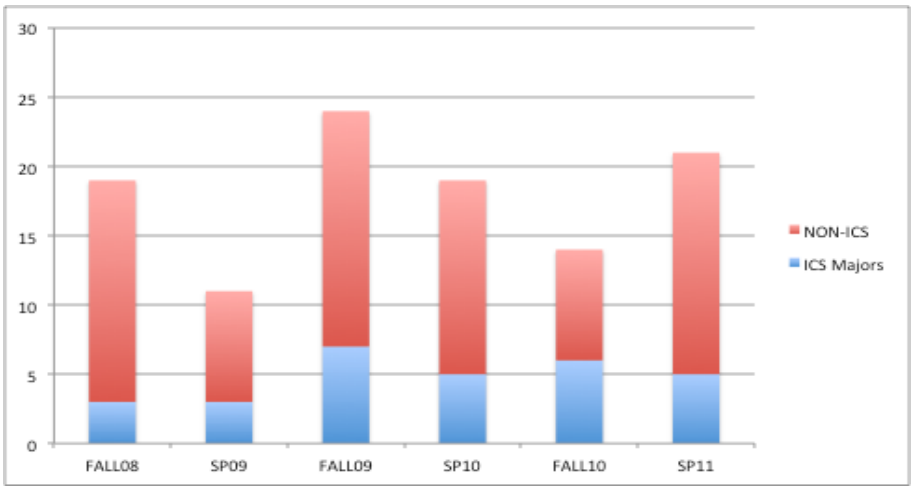


Figure 5. ICS Majors and Non-Majors by Semester

(Note: The information in Figure 5 was extracted from the class rosters and it is not available for public viewing.)

This study takes place in an online course. This course intends to serve students that are not located in the island of Oahu, as well as students that are enrolled in other institutions but have conflicting schedules that prevent them from taking the class. Overtime, this course has become popular with students that simply prefer online learning. Figure 6 shows the student campus' affiliations during the data collection period, spanning from Fall 2008 to the Spring 2011.

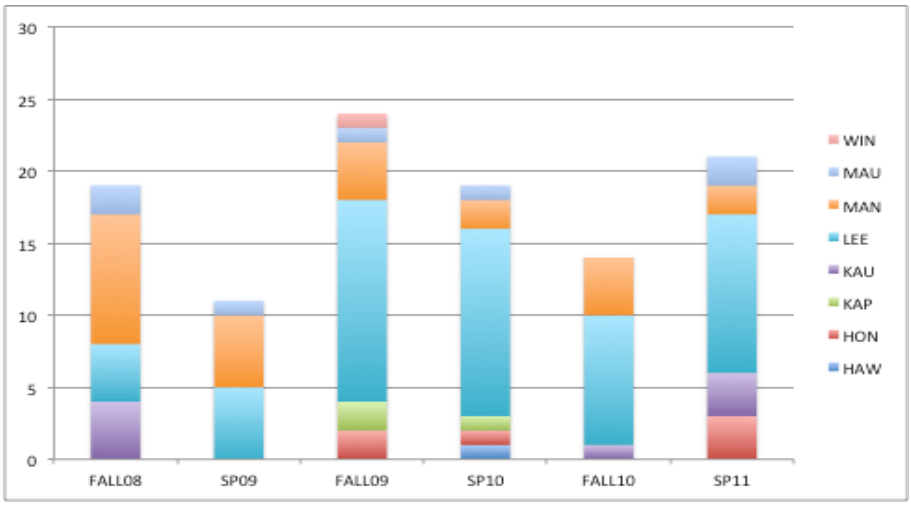


Figure 6. Students by Campus Affiliation

(Note: The information in Figure 6 was extracted from the class rosters and it is not available for public viewing.)

Participants

Every student enrolled in ICS111 online during the Fall2010 and Spring2011 semesters participated in this study. No random selection of any kind took place. The Fall 2010 offering of ICS111 received the SBL-Online treatment while students enrolled during the Spring2011semester remained as the control group. In both semesters, the pre-test, post-test, and a modified version of the Motivated Strategies for Learning Questionnaire (MSLQ) and an end-of-semester questionnaire were administered to students. Students participating in the SBL-Online treatment were also required to participate in SBL-Online sessions as well as to prepare in advance for those sessions. There was no sampling. The treatment given to the students was an integral part of the course.

Participating students received a full explanation on how the course was going to be conducted and they also received proper and timely training on the tools that they were going to be using during the course. This was done for both sections. The tools used for this course are not unique to ICS111 so additional student training was provided by personnel from the Leeward CC Media Center (http://emedia.leeward.hawaii.edu/emc/).

This course begins emphasizing programming constructs and theory, slowly changing focus into the practical aspect of programming. At the beginning of the course, quizzes are given to students almost at the rate of one after each broadcast session to emphasize fundamental theoretical concepts. This goes on during the first third of the course. At that time the emphasis shifts. The amount of quizzes decreases as the amount of time that students are expected to devote to assignments increases. By the end of the semester there are almost no quizzes so that students can focus their attention solely in their programming projects.

This course followed the same rules and standards during the two years preceding this study. The same recorded broadcast lectures have been used every semester. The fact that these courses are structurally identical is the main reason why the data derived from them can be effectively used for the creation of SBL-Online studio sessions.

Settings for This Study

Laulima, the University of Hawai‘i Course Management System

Laulima is a course management system based on Sakai (http://www.sakaiproject.org), currently in use at the University of Hawai‘i campuses for delivering didactic material and assessing students.

Course Components

Courses described in this study gravitate around Laulima. According to this premise, the components for this course can be divided into three categories depending on how they relate to Laulima.

- **Laulima integrated course components.** These components are provided by Laulima and work only within this system.
- **Laulima add-on components.** These components are external but they can be linked to the course management system via web links or other means.
- **Components not related to Laulima.** These components are unrelated to the course management system. They are external to it. They may be mentioned in Laulima, but they cannot be accessed through it.

To access Laulima students are required to have a username and password. This serves as means of official student identification while it provides visitor and access statistics. Laulima also provides privacy for student grades.

With the help of Laulima, students can access course material in a central location. Laulima provides an environment where students can take timed quizzes and tests.

Laulima also provides access to the University's official mail system. Announcements are featured in Laulima's main page. This tool is very useful as a reminder for the different course deadlines.

Laulima Integrated Course Components

Syllabus

The syllabus for ICS111, course structure and grading procedures have undergone negligible changes throughout the lifespan of the online version of this course at Leeward Community College. The following are the course components that prevailed since the Fall 2008. The same weights and components were kept throughout the study.

- Programming assignments 45%

  Thirteen programming , with the same difficulty level.
- Quizzes 20%

  Composed mostly of multiple-choice questions
- Midterm Exams 20% (10% each one)

  Contain multiple-choice questions, open questions and programming is always required.
- Final Exam 15%

  Contains multiple-choice questions, open questions and programming is always required.

Assignments

Assignments are crucial in an introductory programming course, because the nature of the course is primarily to teach students the practical side of programming. Programming is mostly learned by doing. Assignments count for 45% of the grade for

this course, therefore making it the most important element. Assignments, however, cannot be successfully implemented without the theoretical knowledge conveyed during the lecture sessions and reviewed during quizzes.

Assessments

Midterm exams. The Midterm exams given during the four semesters in question have been exactly the same. Midterm exams are listed in Appendix A. Multiple-choice midterm questions are regarded the same as quiz questions (see "Quizzes," below). The data gathered from the programming questions was interpreted the same way as programming assignments and was used as basis for the creation of SBL-Online sessions.

Quizzes. The quizzes given during the four semesters in question have been the same except for the Spring 2009 when thirteen quizzes were administered instead of the usual fourteen. Errors in multiple-choice quiz questions served as basis to prepare the pre-test and post-test that was administered to students as required by the SBL project (CPATH, 2010). The quiz questions with the most errors were the ones used to build the pre-test/post-test. The pretest/posttest was not used in this study.

Class Material

PowerPoint presentation handouts. The course PowerPoint presentations in PDF format are available to students in Laulima. These presentations contain the material used during the pre-recorded course lectures.

To-do lists. Each lecture comes along with activities recommended for students. This list includes suggested readings and suggested practice exercises. To-do lists can be found in Laulima as part of the course resources.

Laulima email. Laulima has its own email system linked to the University of Hawai'i email. Laulima allows every member of the course to email anyone in the class. It displays a list of all the students in class along with their email address. Laulima email facilitates message delivery to students as a group.

Laulima announcements. Announcements may be set up in advance. For these classes whenever a test, quiz or assignment opens, the pre-programmed announcement displays and an email message is sent to all the course participants.

Laulima Add-on Components

Elluminate

The Elluminate tool was provided by Leeward Community College to instructional faculty. There was no choice in the selection of this tool therefore this work does not include a justification for its use. This work is not about the characteristics of Elluminate, instead it describes how it was used. For example, some housekeeping took place at the beginning of the SBL-Online sessions especially during setup time. Students were required to perform an audio test to ensure that they could hear, speak and be heard. Due to technical issues, not all students were able to use all the features that Elluminate had to offer. Chapter 5 describes some contingencies with the medium and how students coped with them.

When logging into Elluminate, students are required to type in their name. Students were instructed in advance to type in some other name or handle if they were concerned about their privacy. Some of them did while others just logged in with their first name.

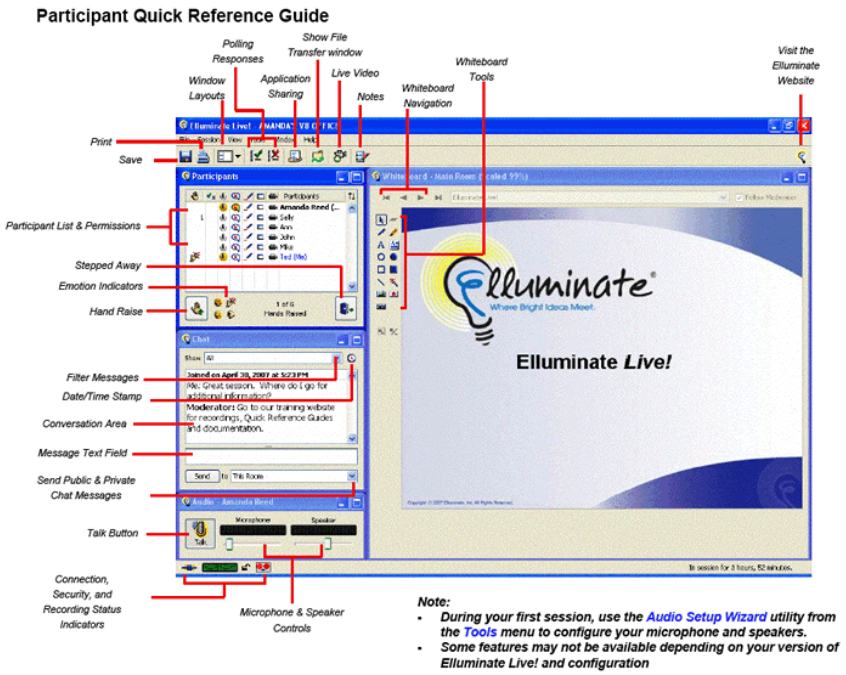


Figure 7. Elluminate Reference Guide

The Elluminate software counts with chat, audio and application sharing and recording capabilities among other things. The Elluminate software recording capabilities allowed SBL-Online sessions to be recorded for later analysis. Students were also able to take advantage of these recordings. They could watch recorded SBL-Online sessions to review course material or recall what was said during these sessions. Students used chat, audio, and application sharing capabilities during the SBL-Online process. As a moderator, the instructor used the same tools and initiated and stopped the recording during each one of the sessions.

To promote a natural behavior, students are told that they will be recorded for their own benefit. They knew that if they what was said or done during the session, they could go back and review the recording. Giving this perspective to students promotes natural behavior "not established or maintained for the sole primary purpose of conducting research" (Hoyle, Harris, & Judd, 2002 p. 362).

Lecture video podcasts

Podcasts contain a downloadable version of the pre-recorded lectures. They are identical to the ones broadcasted on cable TV. Students can watch podcasts in their computers or mobile devices.

Java API

The java application programming interface (API) contains the online java documentation provided by Sun Microsystems, now Oracle. Learning to access and use the Java API is an important skill for ICS students, because it is a real-world reference, constantly updated.

Components External to Laulima

Broadcast lecture sessions

The ICS111 course was taped during the Spring 2008. The lectures in this course have remained the same, except in the Summer 2010 when the first two lectures were re-taped to address changes in the operating system, course setup and procedures. The decision to update these lectures obeyed to changes in the information technology infrastructure within the University of Hawai‘i campuses as well as the adoption of Laulima. The remaining 24 sessions remained the same since they first aired in the Fall of 2008. The static nature of the video lectures and didactic materials affords the assumption that the courses were delivered the exact same way.

UNIX operating system

Unix was selected for this course because it provided homogeneity for assignment development and grading. The Unix server was adopted because even with so many platforms and version of Java, it provided stability. Since assignments were developed in Unix and the process to send them to Laulima from the Unix server is cumbersome. It was established that assignments had to be submitted by email.

Textbook

The textbook used in this course was "Java Software Solutions" by Lewis and Loftus (2008). This book is published by Pearson. In previous semesters, the class used the same book but different editions.

Office hours

Virtual and in-person office hours were offered every semester. Students were also able to contact their instructor via a number of different media. They could email, place a phone call, pay a personal visit or set up a videoconference appointment as needed.

Conventional means of communication

During the semester, students may call the instructor on the phone or visit the instructor in person during office hours. It is unusual for students taking online courses to visit the instructor in person. However, students that also take face-to-face courses on the Leeward Community College campus sometimes visit online instructors.

Institution Review Board (IRB)

The IRB document that covers the SBL project at the University of Hawai‘i, covers this study. Nevertheless, a specific exemption for this study was processed. The Committee of Human Studies (CHS) number for this study is CHS #18439.

The University of Hawai‘i Institute of Human Studies has specific guidelines for instructors that will be doing research involving their own students (http://www.hawaii.edu/irb/). This specific study was exempt because the SBL pedagogy that would be used to conduct the class would have been used regardless. It is considered the same as any other pedagogy that may be selected by the instructor. The population for this study, was self-selected, students chose to enroll in the course. The study took place during two consecutive semesters. In the Fall 2010 students were given the SBL-Online treatment while, students enrolled in the Spring 2011 were the control group.

The ICS111 course follows the ACM curriculum and it is articulated across all the University of Hawai‘i campuses. Therefore, the material taught while this study took place was the same, and it was the same as in the previous semesters.

Following IRB recommendations, students were thoroughly informed about this study. They were told that regardless of their decision to participate in this study, the course activities would remain the same. However, if they decided not to participate, then the data generated by them would not be used. In order to encourage students to participate, extra credit was awarded. For those that did not wish to participate, an option was given to complete a couple of programming assignments for the same amount of extra credit that they would get if they were to participate in the study. One hundred percent of the students in both semesters decided to participate.

Students were informed that the study taking place was for a doctoral dissertation. They were informed that their identities would not be in any way compromised. Student privacy has been thoroughly protected throughout this study.

Students had to officially agree to participate in this study as required by the IRB. Since this is an online course, an online consent form was used. The form can be found in Appendix B.

The Research Paradigm Defined

This study is twofold. First it compares two implementations of the same online course, the traditional, vs the one where SBL-Online sessions are implemented. In order to compare the two courses, this study uses the modified MSLQ, the end-of-semester questionnaire, and course grades. These are precisely the elements that the SBL-Project requires. Secondly, this study analyzes the inner workings of SBL-Online sessions. Session recordings are transcribed and coded. Results from both parts of this study, are supported and framed by the thick data collection taken place throughout the process. Figure 8 shows the research paradigm of this project.

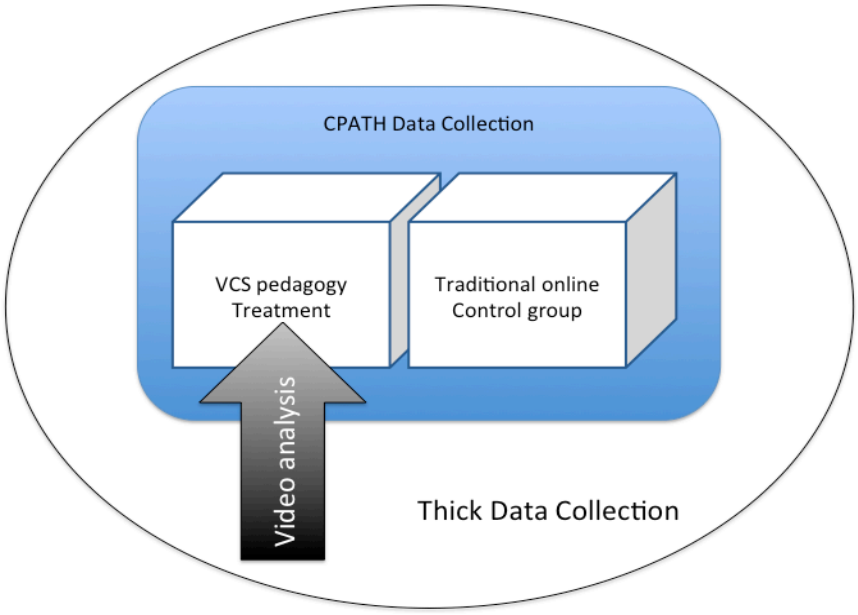


Figure 8. Research Paradigm

While the base for these courses in general is the Laulima course management system and its elements as described before, the setting for the SBL-Online is Elluminate. Elluminate is a natural context-rich setting not created for research purposes. The Elluminate virtual classroom played the role of the "meeting place" for SBL-Online sessions. Data gathered in Elluminate will be examined from a holistic perspective. As the instructor and as a researcher, I assumed the role of participant when interacting with my students during each SBL-Online implementation. Persistent observation was achieved through the recording of SBL-Online sessions.

An audit trail has been established by keeping all the raw data collected for this study. It includes all the SBL project-related data as well as video recordings and transcripts. The Laulima course management provides access to past courses. The data obtained from it, remains stored within and can still be accessed at any time.

Procedures for Creating, Coding, and Analyzing Data

A great amount of data was collected during the two semesters in which the study took place. The data inventory showcased in Figure 9 shows the data that was collected,

and operationalized in order to answer the research questions. The operationalization displayed within Figure 9 is the beginning pathway to a minimum degree of analysis. However, analysis options remained open for the duration of the study.

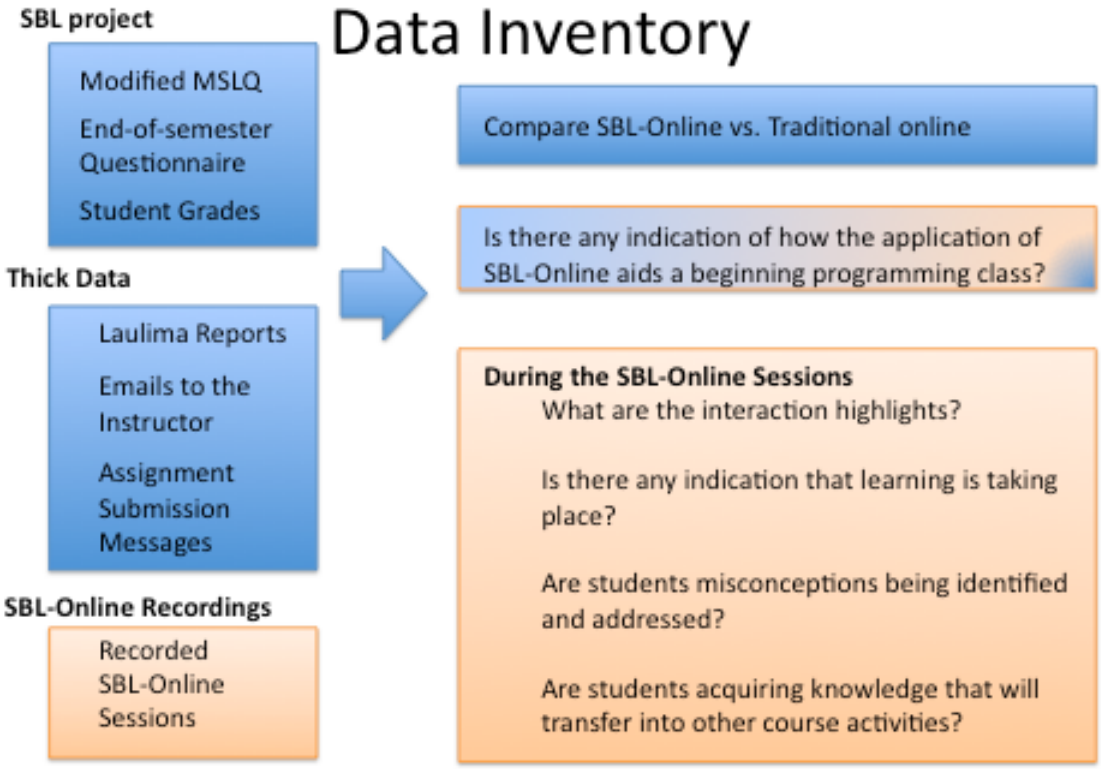


Figure 9. Data Inventory

Student interaction during SBL-Online sessions will answer the questions regarding interaction highlights, misconceptions and transfer of knowledge. Once this study is complete, it will determine if there is any indication of how the application of SBL-Online aids a beginning programming class. The way that this study was structured allowed for discovery and exploration.

Instrumentation

This section describes the instruments used for gathering data. All these instruments can be found in Appendix B. For a quick overview of the instruments and their role in this study please refer to Figure 9.

The Motivated Strategies for Learning Questionnaire (MSLQ)

A subset of MSLQ was given to the students at the end of the semester as part of the SBL project. The MSLQ was designed to measure college students' motivational orientations and their use of different learning strategies. Researchers perceive the MSLQ as a reliable and useful tool that can be adapted for a number of different purposes (Duncan & McKeachie, 2005). The MSLQ was developed by Bill McKeachie and Paul Pintrich at the University of Michigan in the early 80s. It took them over ten years to complete the process (Artino, n.d.). Pintrich, Garcia, Smith, and McKeachie (1993, p. 801) explain the purpose of the MSLQ:

> was designed to assess motivation and use of learning strategies by college students. The motivation scales tap into three broad areas: (1) value (intrinsic and extrinsic goal orientation, task value), (2) expectancy (control beliefs about learning, self-efficacy); and (3) affect (test anxiety). The learning strategies section is comprised of nine scales which can be distinguished as cognitive, metacognitive, and resource management strategies. The cognitive strategies scales include (a) rehearsal, (b) elaboration, (c) organization, and (d) critical thinking. Metacognitive strategies are assessed by one large scale that includes planning, monitoring, and regulating strategies. Resource management strategies include (a) managing time and study environment; (b) effort management, (c) peer learning, and (d) help-seeking. Scale reliabilities are robust, and confirmatory factor analyses demonstrated good factor structure. In addition, the instrument shows reasonable predictive validity to the actual course performance of students.

Overall the MSLQ appears to be a sound instrument except for a small flaw with respect to its internal validity in three of the MSLQ sub-scales. This obeys in part to the small number of items within them. However, the remaining twelve subscales have an alpha coefficient 0.74 and above.

The modified MSLQ helps assess student attitudes and motivation and it focuses on the following MSLQ subscales: intrinsic motivation, extrinsic motivation, self-efficacy,

peer learning, critical thinking, self-regulation, and sense of community. Analysis of the answers to these questionnaires was performed by the SBL project team, and processed again for this study.

End-of-Semester Questionnaire

The end-of semester questionnaire gathers data about what students think about the way the course was conducted. It includes questions about teamwork, comfort, confidence, quality of feedback received, etc. This questionnaire is an excellent way to wrap up the SBL-Online experience while providing an insight into the thoughts of students in the control group. The end-of-semester questionnaire can be found in Appendix B.

Inside SBL-Online

Every SBL-Online session was recorded. SBL-Online sessions took place online using the Elluminate software. When launched, Elluminate automatically offers to record any session taking place. Elluminate has its own consent form for students to agree to be recorded, otherwise they cannot proceed to the Elluminate environment. This consent is independent from the consent given to students participating in this study. This particular consent is part of Elluminate's legal disclaimers. Elluminate recordings for this course did not show student's faces, only their voices, chat messages and anything that was displayed or shared in the Elluminate screen. This provides a high degree of privacy for the students and a perfectly good video to be analyzed.

In order to analyze these videos, several codebooks were revised. Two of them are product of the SBL project Hundhausen, Fairbrother and Petre (2011) and Sheperd-Jones (2011). However these codebooks are very specific for human-computer interaction and design courses, which are radically different from programming courses. The codebook that can be better used as a base to code SBL-Online sessions is by Curtis and Lawson (2001) and it is shown in Figure 10.

Codes	Description	Example from an SBL activity
GS	Group skills: a generic code applied to expressions that encourage group activity and cohesiveness	I know that [names] have given you good advice, but I think that is worth knowing that you need patience
OW	Organizing Work: Planning group work; sharing set tasks and details	Who is going to fill out the form in Laulima?
IA	Initiating activities; Setting up activities such as chat sessions to discuss the progress and organization of group work	Who wants to meet to work on the assignment?
HeG	Help giving; Responding to questions and requests from others	To access the chat room, click on virtual chat in the blackboard; chat screen will come up, click on enter.
FBG	Feedback giving; providing feedback on proposals from others.	If you do not align the braces you are more prone to making mistakes
RI	Exchange resources and information to assist other group members	You just need to do _____
SK	Sharing knowledge: Sharing existing knowledge and information with others	I think we also need to give thought to the following. 1. The issues of quality/efficiency in teaching and learning.
Ch	Challenging others; challenging the contributions of other members and seeking to engage in debate	
Ex	Explaining or ellaborating	the way that it works is by …
HeS	Help seeking; seeking assistance from others	Can anyone tell me how to do ….this?
FBS	Feedback seeking; seeking feedback to a position advanced	What is your opinion on this? Is it done correctly.
Ef	Advocating effort; using others to contribute to the group effort.	Haven't heard from you are you having technical difficulties? Are you ok?
ME	Monitoring group effort; comments about the group processes and achievements	Shall we move to the next question? I think we are done with this one.
RM	Reflecting on medium; comments about the effectiveness of the medium in supporting group activities	In the chat: My microphone is not working so I will be using the chat
SI	Social interacton; conversation about social matters that are unrelated to the group task. This activity helps to break the ice	My weekend is pretty hectic I have my family is flying in from Mexico…

Figure 10. Coding Categories by Curtis and Lawson (2001)

The Proactive Instructor

As part of the preparation and settings for this study data from past courses was collected. The courses in question span from the Fall 2008 to the Spring 2010. This collection included student demographics, grades and errors made by students in assignments and assessments given to them during the course of the semester.

Collecting Grades

Laulima is a course management system used by most distance education courses in the University of Hawai‘i campuses. This system, by default archives past courses, making them available for faculty through the years. This capability allowed me to go back and collect past data from the ICS111 courses that I had previously taught. The following is a list of data that was collected.

- Grades for tests, quizzes and assignments
- Imperfect tests, quizzes and assignments submissions
- Student campus affiliations
- Final grades

This data was used as a guide for the creation of this project. Specifically to find out where in time and in which activities students made the most errors.

Error Categorization and Descriptions

In order to quantify errors it is necessary to create categories. The error category selection process was described in chapter 3. After careful revision of this error list, the error categories described by Robins et al. (2006) was selected for this study. The categories have been broken down to address four different kinds of errors that may occur; background, general, language specific and submission setbacks.

B	**Background Errors**
B1	Tools, operating system issues, lost files, web-posting issues, file permissions.
B2	Understanding the task, students do not understand what to do
B3	Stuck on program design, students know what to do but they don't know how to do it

Figure 11. Errors Related to Programming Tools and Task Understanding

Figure 11 lists the possible background errors, related to programming tools and understanding, which may happen due to problems with the programming environment. They may also be caused by inexperience with the tools needed to accomplish programming assignments or the misunderstanding of the task at hand.

G	**General Programming Errors**
G1	Problems with basic structure, code outside methods, data fields outside the classes, etc.
G2	Problems with basic object concepts such as creating and using instance objects.
G3	Problem naming variables, constant or methods suggesting that they do not understand their purpose
G4	Problems with trivial mechanics, braces, brackets, semi-colons, typos, spelling, naming conventions, import statements forgotten, output formatting, the use of magic numbers.

Figure 12. General and Structural Programming Errors

General errors are non-specific to the programming language used. General errors include typos, spelling mistakes, naming conventions, problems with creating and using

objects, having mismatched braces etc. General errors can happen at any time during the semester.

Experience has shown that errors as those described in Figure 12 occur mostly at the beginning of the semester, because students are not yet familiar with syntax rules and programming constructs. Later on during the semester, students know the programming rules but distractions may cause these errors to continue to appear. General errors happen throughout the whole semester therefore they cannot be addressed specifically within any particular assignment.

Figure 13 displays the topics that may cause Java-specific errors in a well-modularized medium-granularity categorization. These errors are the ones that can be provoked and targeted within particular assignments. These are the ones that will be addressed via SBL-Online sessions. These errors may be generated by both the inappropriate use of programming tools or a mistake in the design of the solution to a problem.

S	Topics That Cause Errors Specific to the Java Programming Language
S1	Control flow, basic sequential flow of control. The role of the main and init methods. Call and return of methods.
S2	Loops, related to repetition, loop headers, bodies and blocks
S3	Selection, if-else and switch, blocks related to selection and validation.
S4	Booleans and conditions, formulating and understanding Boolean expressions (doesn't include precedence)
S5	Exceptions, throw/catch
S6	Method signatures and overloading
S7	Data flow and method header mechanics includes arguments, parameters and return-type values. Mismatching parameter specs, incorrect return values, incorrect use of void.
S8	Terminal or file I/O.
S9	Strings and string methods
S10	Arrays as data structures, subscripts, contents, declaration and initialization. It includes ArrayLists.
S11	Variable initialization and assignment
S12	Visibility and scope, reuse of identifiers
S13	Expressions and calculations, precedence and random number generation and use
S14	Data types and casting
S15	Reference types, the behavior of reference types being different from that of primitive data types.
S16	Class vs instance, the use of "static"
S17	Accessors and modifier methods their purpose and use
S18	Constructors concept and purpose
S19	Hierarchies, inheritance, extends, shadowing, super
S20	GUI mechanics, action performed and action listeners.
S21	Event-driven programming, flow of control in an applet.

Figure 13. Errors Specific to the Java Programming Language

The error categorization from Robins et al. (2006) does not address issues with incomplete code, code that does not compile or late work. These three categories were appended to the adopted error list. Figure14 contains issues particular to assignment delivery. Once the assignment delivery setbacks were appended to the adopted error category, the categorization was considered ready to use by this study.

	Assignment Delivery Setbacks
I	Incomplete code
NC	Complete code doesn't compile
LW	Late work

Figure 14. Assignment Delivery Setbacks

Errors as a Guiding Light

The SBL-Online differs from others studio-based learning implementations in that its design is based on evidence left behind by students who previously took the CS1 course. Data from assessments and assignments from the four semesters preceding this study (Fall 2008, Spring 2009, Fall 2009 and Spring 2010) was collected and analyzed. This was possible because the ICS111 course has been virtually identical every semester from the fall 2008 to the Spring 2010 from the practical and conceptual points of view. The material covered during the broadcast lectures has been exactly the same and quizzes and midterms have been identical. A few differences have taken place within assignments. However, even when the assignments have not been exactly the same, they require that students perform similar calculations and the logical and critical thinking required to successfully solve them is the same.

Beginning with the Fall 2008 semester, every assignment that did not received full credit was revisited in order to extract and categorize student errors. The total students enrolled in this course through this four-semester period were 73. Not all students turned in all the assignments. Table 1 shows student assignment and error distribution throughout the semesters.

Table 1. Assignment Turnout During the Fall 2008–Spring 2010 Period

Semester	Students enrolled	Assignments submitted	Assignments with errors
Fall 2008	19	223	94
Spring 2009	11	124	41
Fall 2009	24	255	81
Spring 2010	19	178	67
Total	73	780	283

Midterms were also part of this error categorization. Two midterm exams were given every semester. The first midterm contained two programming questions while the second midterm contained three programming problems and one open programming question. Just as with assignments errors were categorized, while questions that received full credit were not revisited. Please refer to Appendix A for midterm exam questions.

Table 2. Summary of Errors During Midterm Exams

Results outcomes of programming questions by semester		Midterm 1 programming question 1	Midterm 1 programming question 2	Midterm 2 open question	Midterm 2 programming question 1	Midterm 2 programming question 2	Midterm 2 programming question 3	Totals
Fall 2008	Submissions with imperfect scores	5	10	3	9	5	1	33
	Total submissions	17	17	14	14	14	14	90
Spring 2009	Submissions with imperfect scores	3	5	2	4	3	1	18
	Total submissions	10	11	6	8	8	9	52
Fall 2009	Submissions with imperfect scores	12	6	13	13	10	5	59
	Total submissions	23	23	21	19	19	19	124
Spring 2010	Submissions with imperfect scores	6	9	3	6	5	6	35
	Total submissions	18	17	13	12	13	12	85

As shown in Table 2, a total of 145 incorrect midterm-programming questions were revisited and the errors categorized. Once both the assignment and programming errors were categorized the results were sorted by error-count. Table 3 shows the results. Not all the error categories are listed. Those categories with no errors found during this data collection were omitted from the table.

Not surprisingly the findings of this study coincide with results presented by (Robins et al., 2006) as well as results of others cited by them. Issues related to general and background problems figure among the top five, in this case surpassed only by errors in arithmetic calculations and precedence. Results coincide with my experience as a computer science professor where I have seen many of my students stuck with general and background errors derived from mismatched braces, typos and missing or duplicate variable declarations.

Table 3. ICS111 Programming Errors Listed in Descending Order

Category	Test	Assignment	Total	Topics Causing Errors
S13	66	28	94	Expressions and calculations, precedence and random number generation and use
B2	19	44	63	Understanding the task.
S3	12	46	58	Selection, if-else and switch, blocks related to selection and validation.
B3	15	40	55	Stuck on program design, students know what to do but they don't know how to do it
G4	8	42	50	Trivial mechanics, braces, brackets, semi-colons, typos, spelling, naming conventions, import statements forgotten, output formatting, the use of magic numbers.
S4	17	15	32	Booleans and conditions, formulating and understanding Boolean expressions (doesn't include precedence)
S20	21	6	27	GUI mechanics, action performed and action listeners.
S5	5	24	26	Exceptions, throw/catch
S2	15	7	22	Loops, related to repetition, loop headers, bodies and blocks
I	8	13	21	Incomplete code, it lacks at least half of the required functionality.
NC	NA	20	20	Code doesn't compile but the code shows that more than half of the functionality has been implemented.
S14	0	10	10	Data types and casting
S1	2	6	8	Control flow, basic sequential flow of control. The role of the main and init methods. Call and return of methods.
B1	0	7	7	Tools, Operating system issues, lost files, web-posting issues, file permissions.
S9	0	7	7	Strings and string methods
S10	0	7	7	Arrays as data structures, subscripts, contents, declaration and initialization. It includes ArrayLists.
S7	0	4	4	Data flow and method header mechanics includes arguments, parameters and return-type values. Mismatching parameter specs, incorrect return values, incorrect use of void.
S17	0	4	4	Accessors and modifier methods their purpose and use
S21	0	4	4	Event-driven programming, flow of control in an applet.
S12	0	3	3	Visibility and scope, reuse of identifiers
S18	0	2	2	Constructors concept and purpose
G1	0	2	2	Problems with basic structure, code outside methods, data fields outside the classes, etc.
S8	0	2	2	Terminal or file I/O.
LW	N. A	2	2	Late work submission.
G3	0	1	1	Problem naming variables, constant or methods suggesting that they do not understand their purpose
S15	0	1	1	Reference types, the behavior of reference types being different from that of primitive data types.

Building and Placing SBL-Online Sessions

Once the errors had been categorized, the next step was to find the points in the curriculum where the confusing concepts were introduced so that they could be immediately reinforced. The problem is that most of the errors fall in the background and general categories. Even when these errors happen throughout the semester, they are for the most part covered during the first few sessions of the course. This kind of errors is too general and can't be directly addressed with specific assignments. Every one of the assignments given in this course will require students to use general and background knowledge to complete them. Given this fact SBL assignments were created to address the errors found in the "specific to Java topics" category, which in turn will require general and background knowledge to be implemented.

Error Count	Error Description		Covered in lecture (s)
94	S13	Expressions and calculations, precedence and random number generation and use	5, 6, 7 and 9
58	S3	Selection, if-else and switch, blocks related to selection and validation.	13 and 14
32	S4	Booleans and conditions, formulating and understanding Boolean expressions (doesn't include precedence)	5, 13, 14 and 16
27	S20	GUI mechanics, action performed and action listeners.	15, 17 and 18
26	S5	Exceptions, throw/catch	10, 12 and 21
22	S2	Loops, related to repetition, loop headers, bodies and blocks	16

Figure 15. The Most Common ICS111 Errors and the Lecture Where the Related Material Is First Introduced

Evidence from Figure 15 suggests that, no SBL session should take place before lecture five. In order to find the best placement for the SBL assignments, the lectures that cover the troublesome material are sorted in broadcast order. The order of these sessions can be seen in Figure 16. Lectures have been grouped to form three studio-based learning sessions. This is the amount of sessions recommended by the SBL project (CPATH, 2010).

SBL	SBL Session 1						SBL Session 2					SBL Session 3
Sets	Set1			Set2								
Lecture	5	6	7	10	12	13	14	15	16	17	18	21

Figure 16. Lectures That Cover Troublesome Material in Chronological Order

The optimal SBL session designed specifically for this study should have the following characteristics:

1. Address topics considered critical; Critical topics are that were derived from error analysis from past semester information
2. Provide opportunities for creativity. Assignments should not be extremely difficult but they should require a considerable amount of coding (Woodley & Kamin, 2007).
3. Assign projects at critical points on the curriculum. These points are indicated by the concentration of critical errors.

According to Table 3, the way that SBL-Online assignments are designed, they cover 80.2% of the most common errors.

SBL-Online Session 1

The first group of lectures containing troublesome material encompasses lectures five to seven. During these lectures the basic language constructs are introduced. In lecture five, numeric operations are explained and precedence is discussed. Lecture six is about the use of constants and different numeric data types as well as to strongly recommend the avoidance of magic numbers. Lecture seven wraps-up this stretch by explaining how different numeric data types can be mixed and how to format numerical output. Even when apparently this is a good place to assign the first studio session an assignment given at this point with the students' limited skill set will not comply with requirement two of the optimal SBL session required characteristics for this study. Therefore the first set will be further expanded to include the second set to comply with this requirement.

The first studio assignment is given after session 13 when students already know how to handle possible input exceptions and can use if-statements to validate information. BMI.java is the first assignment that not just has the necessary characteristics to be catalogued as a studio assignment but it also covers the first set of topics found to be "complicated" for students taking this course. BMI.java assignment requirements can be found in Appendix C.

In set 2, student programs become interactive, and allow more variants in student's possible solutions. Lecture ten introduces student to user input, while, lecture twelve targets Java exceptions. When moving into lecture 13, students are introduced to conditions. It is at this point when the troublesome material be tested and reinforced. At this point students are expected to have enough knowledge to take over larger projects and enough confidence in their abilities to present their code to their peers. It is at this point that for the first time the three required SBL requirements are present.

SBL-Online Session 2

This second session will target random number generation, GUI mechanics, flow of control, posting applets and event-driven programming, which account for a big part of the overall errors. Lecture 14 covers switch statements, while lecture 16 covers loops. Lectures 15, 17 and 18 deal with applets and Java on the web. After those four lectures it is time to implement another studio session. The ideal timing for another SBL-Online

session to be assigned is right after session 18 and will force students to implement concepts learned thru the last four sessions sprinkled with material from previous sessions.

The SBL second selected assignment is SlotMachine.java. This assignment, addresses a second set of complex topics for ICS111 students and it also features the required studio assignment characteristics. SlotMachine.java assignment requirements can be found in Appendix C.

SBL-Online Session 3

The last target problem is located at session 21, where exception generation and throwing are introduced. The topic of exceptions usually involves reading and filtering user input as well as validating data before creating objects. In order to fully address exception handling, it is necessary to create a studio assignment that will require exception handling, as well as the creation of customized exceptions.

To create customized exceptions it is necessary a significantly large assignment. An assignment this long is usually given in two parts. The first part is relatively easy and very similar to the process of following a recipe, where students make little or no mistakes. It consists of the creation of the custom exception and the object classes. The coding of these classes gives no room for creativity and the driver class is provided.

The second part involves the creation of a driver class involving the reading and validation of user input. This part is the one used as SBL session three. This assignment provides opportunity for creativity and mistakes alike. This is a critical point for the third studio session to take place. It will allow students to use most of the material that they have learned during the duration of the semester with emphasis in selection statements, loops, arrays as well as object creation and exceptions.

The third selected assignment is CDStore.java. Please refer to Appendix C for a full description of this assignment.

SBL Project Procedures Within SBL-Online

The data collection requested by the SBL project was done following their provided schedule. Most of the SBL project participants are distributed around the United States therefore data collection was done with online. Appendix B contains the instruments to gather data.

The ICS111 course organization was divided into 3 parts to better fit this study.

The first part took place before the semester began and spanned throughout the first two weeks of classes. During this period, participants were identified, and notified about this project. A URL with the consent form was given to the students so that they could accept or decline to participate in this study.

The second part encompasses most of the semester. It is where the SBL sessions take place and most of the practical data is generated and collected. The third and last part spanned through the last week of the semester and the week after. It is here where grades were assigned and data collection was inspected for compliance with both the SBL project and this study.

This study took place during a full year. During the Fall 2010 semester, the course featured the SBL-Online pedagogy. During the Spring 2011 semester, the course ran without the SBL-Online sessions. This allowed for direct comparison between performances of both student populations. Both courses included the same assignments, quizzes, midterms and final exam. All the activities for both semesters were distributed equally throughout the semester. Figures 17, 18 and 19 show the chronological sequence for both semesters. Because different semesters have different holidays and durations, the semester chronology is based solely on when the broadcast sessions air.

SBL project activities during the first week of classes • Knowledge assessment (pretest)				
Session	Topics	No SBL Project activities after the first week of classes	Quiz (material)	Assignment
1	Syllabus		Q01 (Session 1)	
2	UNIX overview		Q02 (Session 2)	
3	First Program		Q03 (Session 3)	A01-Poem
4	Escape Sequences and variable names		Q04(Session 4)	A02-Escape Characters
5	Primitive data types and relational operators		Q05(Session 5)	
6	Working with numbers		Q06(Session 6)	
7	Mixing numbers		Q07(Session 7)	A03-Tip Calculation
8	String methods		Q08(Session 8)	A04-Phrase
9	Strings and Random		Q09(Session 9)	A05-Dice
10	User Input			
11	Midterm 1 Review		Midterm 1 (sessions 1 to 11)	

Figure 17. Semester Chronology of Events Based on Broadcast Sessions, Part 1

The first lectures of the course are the only ones that directly address background errors B1, B2 and B3 as well as general errors G1, G3 and G4. To find these errors within programming assignments is extremely hard and time-consuming. The frustration experienced finding and fixing trivial errors prompts students to consider the possibility of dropping out of CS1 courses (Kinnunen and Malmi 2006).

The first part of the course featured in Figure 18 focuses mostly on programming theory and provides students with the tools and background information needed to get the job done. It features almost one quiz after each session with the purpose of reinforcing java basic programming structures and the programming process. The first two assignments are very similar. The goal is that students will become familiar with the Unix operating system and the process of writing programs as well as submitting them via email. Assignment three deals with numbers and calculations. Assignment four is about string manipulation, while assignment five introduces students to random number generation and its implementation to create games. Up to this point, students are writing programs that do not require any user input.

The second part of the proposed study begins with the broadcast of session 12. At this point in the semester, students are already able to get input from users therefore they need to validate the information received in order to adequately process it. This part of the course features less quizzes and more complex assignments.

Session	Topics	SBL Project Activities	Assessment (Material)	Assignment
12	Exceptions			
13	What-if?	SBL-Online Session 1 (A06)	Q10 (Sessions 12 and 13)	A06-BMI Calculation
14	Comparing			
15	Java on the Web		Q11 (Session 15)	
16	Loops		Q12 (Session 16)	A07-Drawing Applet
17	GUI Elements			
18	The Interactive GUI	SBL-Online Session 2 (A08)	Q13 (Sessions 17 and 18)	A08-Game Applet
19	Midterm 2 Review		Midterm 2 (sessions 12 to 19)	

Figure 18. Semester Chronology of Events Based on Broadcast Sessions, Part 2

The last part of the proposed study begins after the second midterm. With only one quiz and three assignments is by far the most involved part of the semester. At this point the only area that still needs addressing through SBL projects are exceptions.

Session	Topics	SBL Project Activities	Assessment (Material)	Assignment
20	Objects			
21	Exceptions – Creating your own		Q14 (Sessions 20 and 21)	A09-CD class
22	Arrays			A10-CD Exception
23	Static Methods			
24	ArrayLists and Wrapper classes			A11-Array of Grades
25	Files	SBL-Online Session 3 (A12)		A12-Array of CDs
26	Final Exam Review			

Figure 19. Semester Chronology of Events Based on Broadcast Sessions, Part 3

Describing the SBL-Online Studio Session

SBL-Online sessions consist of two parts. The first part is the same as in any regular homework assignment. For this assignment instructions along with its grading rubric were handed to students. The second part included attending the SBL-Online studio session and the assignment resubmission.

Assignment submission

Students were required to submit their assignment via email to an account set especially for this. Upon submission, students got an automatic "receipt" indicating that their assignment was successfully submitted for grading. The sole purpose of this digital receipt is to give students peace of mind.

The instructor then grades the assignments and provides students with feedback within a week. Preparations done immediately before the SBL-Online sessions

After students received their graded assignments and feedback they were assigned into a team to work on for the second part of the process. (Whittington & Bills, 2004), explain the benefits of grouping students by level-of-knowledge in order to help them learn the class material better. When structuring student teams for code review, (Hundhausen, Agrawal et al., 2010) followed the same principle. They created peer review teams where the students were at the same programming level. For this study,

students were ranked and assigned to teams. Student rankings were done according to the grades that students had up to the time that the ranking took place.

SBL-Online teams should count with least three members. One member will be the author. The author is the team member who wrote the code that was distributed before the SBL-Online session. During the SBL-Online, the author will be presenting the code to the rest of the team. A second team member will be the recorder. This person will take notes during the SBL-Online session. Besides these two members there may be one or more members who will act as code-reviewers.

There were three SBL-Online sessions. For each one of these sessions, students were ranked and then assigned into teams. Students were assigned to different teams during the course of the semester. Once students were assigned to teams they were also assigned a role within the team. Roles were rotated. The intention was that each student would have at least one opportunity to get his/her code reviewed during the semester.

Students' responsibilities

In preparation for the studio session, the code to be discussed should be distributed among team members at least a day in advance. Before the studio session, students were required to review the code and fill out an individual code inspection form.

To begin the studio session, the author must explain the code. Once the author's code presentation is over, every member of the team is given the opportunity to express an opinion about it. The author is then supposed to sit back and observe the review process without interrupting unless reviewers ask questions. This was not the case. Once the studio process is complete, the recorder is required to fill out and submit one group inspection form. This form should be a compilation of the members' thoughts and consensus as a group. A studio-based-learning group inspection form can be found in Appendix C.

SBL-Online sessions took take place over Elluminate Live!. The Elluminate Live! Counts with private and public chat, polling, quizzing, emoticons, whiteboard, voice over IP, application sharing, file transfer and webcam tools among others. Elluminate is provided by Leeward Community College to improve online teaching as well as to enhance face-to-face courses, therefore, students may have used Elluminate prior to taking this course.

Inspection forms

The individual and the group inspection forms are the same. The process of filling them up is what makes them different. Each student fills out individual forms independently before the studio session. Group forms, on the other hand, are filled by the team during the studio session. The group inspection form is intended to be a consensus code evaluation that has been discussed and agreed upon by all team members.

During the SBL-Online

When students are doing teamwork, instructors usually go around the classroom checking on each team, commenting in their work and asking if they have any questions. During the SBL online sessions there is no classroom, but instead the Elluminate virtual room. In this environment, the instructor can get away from the computer and come back online approximately every 15 min to mimic the classroom experience.

Students interact in Elluminate unsupervised. Technical are usually resolved before the session begins. Session recording may be the only element that may be deemed "unnatural," when students listen to a message from the Elluminate software indicating that the recording has started. After this, students perform the code review in their own. At the end of the session, when the recording is turned off, Elluminate will speak the message "recording stopped."

Elluminate records the sessions without the need for a webcam. Since there is no camera present, they tend to act "normal." Elluminate recordings capture only the screen where the code being reviewed is displayed. Any cursor movements are recorded. Verbal utterances and public chat messages by SBL-Online participants are recorded as well.

An extra role within Elluminate

The SBL-Online Elluminate-based online implementation varied slightly from traditional studio-based learning implementations. In this case one team member takes on an additional role as an Elluminate moderator. Elluminate moderators can grant or revoke privileges from session participants, these privileges include the microphone, camera and application sharing privileges among others. Application sharing is the most important of these privileges because it allows the author to share and explain the code with the team.

Reflections About Validity and Reliability

The material given to students is exactly the same in the traditional and in the SBL-Online section. Both are based on common semester timelines. Since students attitudes have the potential to make course experiences different, the lectures were pre-recorded and the explanations remained the same semester after semester, therefore students were exposed to the exact same material. Students attitudes have the potential to make the course experiences different. Some students naturally communicate more often with the instructor or visit the instructor in person while others remain silent. These are the kinds of factors that cannot be controlled or documented.

Limitations of This Study

- This study uses observation and analysis. Inherent limitations of this are that it is only possible to show that events are related but it is not possible to determine the cause and effect of a variable over the population.
- A biased issue may arise due to student self-selection. Students electing to take this class online may be good time managers. They may be better time managers than the rest of the student population.
- It was not possible to control personal face-to-face interaction among students in real life while they are enrolled in the class.
- One possible confounding variable when comparing SBL-Online vs the traditional online implementation of the course is that each studio session represents an extra activity that the traditional course does not have. It may be argued that because students spent extra time doing additional work, this by itself may result in better grades regardless of the studio. Video analysis, MSLQ findings and the end-of-semester questionnaire help explain how studios impacted student's learning habits.
- Social interaction validity threat is considered non-significant because the treatment is not administered in the same semester so it is less likely that students will try to imitate what other students are doing.

Summary

This chapter has explained in detail the process for setting up SBL-Online within a CS1 course as well as the settings for the control group. It described the population for this study as well as the tools used throughout the study. This chapter enumerated the activities taking place during the semester as well as the different roles students undertook as they participated in these courses. Different tools used and their characteristics were described. Lastly, the perceived limitations of this study are listed.

Chapter 5. The Presentation of Data

Introduction

This chapter presents the data collected during the SBL-Online study. It focuses on addressing the research questions, therefore it begins by explaining how each question will be addressed with the data collected. It continues describing the data collected, emphasizing the outcomes and relating them to the research questions.

Design and Methodology of This Study

Part of the SBL project methodology was directly applied towards this study. The SBL project encompasses the application and analysis of the modified MSLQ, student grades, and the end-of-semester questionnaire. This study supplements the SBL project approach with SBL-Online sessions video recordings as well as thick data collection including grades, assignment solutions, assessments, visits to Laulima, student emails and assignment submission messages. Semester grades and assignment submission messages can be found in Appendix E.

The Purpose of This Study

The purpose of this project is to identify "learning moments" during the SBL sessions in order to describe how SBL-Online impacts student's behavior, attitude towards the course and semester grades in an online CS1 course.

In order to address the duality of the study, the data gathered during this research project is presented in two different ways. The first part looks to identify learning moments during the studio session. This part refers to the inner workings of SBL-Online sessions. The second part refers to the impact of SBL-Online in different course outcomes. This part is heavily related to the SBL project as detailed in chapter three, and justifiably, it adopted the SBL project methodologies and procedures.

The first part of this presentation of data will focus on the inner workings of SBL-Online sessions and any immediate effects produced. This part of the data will present an inner view into the mechanics of SBL-Online.

The SBL project has systematically compared SBL implementations vs. traditional implementations of the same course over the years. Therefore the second part of the study, addresses the differences between the two incarnations of the ICS111 course. The Fall 2010 semester depicted an environment in which the SBL-Online was implemented while the Spring 2011 semester featured a traditional online embodiment of the same course.

Data gathered during this study are expected to answer each one of the research questions, as shown in the data analysis in chapter three, as well as the data inventory shown in Figure 9.

Research Questions, and How They Are Addressed

1. **Is there any indication of how the application of SBL-Online aids a beginning programming class?**

   To answer this question it was necessary to look at quizzes, test results, assignments as well as other indirect evidence of student success, such as student's emails and assignment submission messages.

   During the SBL-Online Sessions:

2. **What are the interaction highlights?**

   There was an intense review of student-instructor interaction, student-student interaction and student content interaction in and as much depth as the thick data collection allowed. These interaction highlights are showcased within the SBL-Online session highlights. They were complemented with student emails and the end-of-semester questionnaire.

3. **Is there any indication that learning is taking place?**
   This has been addressed by analyzing SBL-Online session videos and transcripts, as well as student work including programing projects and assessments. This information was complemented with the end-of-semester questionnaire.

4. **Are students' misconceptions being identified and addressed?**
   When misconceptions arise they usually emerge via email directly addressed to the instructor or during SBL-Online sessions while students interact with their peers.

5. **Are students acquiring knowledge that will transfer into other course activities?**
   Knowledge can be transferred directly from the SBL-Online session into the assignment resubmission. This will be the most significant transfer, and the one that can be more precisely quantified.

Data Creation and Collection

Data collection took place during the spring 2010 and fall 2011 semesters Figure 19 shows the chronology of events during the two semesters when this study took place. The timing of events within the semester is an integral part of the settings. Therefore the best part to present the data collected is using a timeline, in which the events are triggered by the broadcast of a lecture.

Session	Topics	Data Collected	Assessment (Material)	Assignment
1	Syllabus	Modified MSLQ Throughout the semester, all quizzes, assignments and tests are collected. Emails sent to the instructor are collected.	Q01 (Lecture 1)	
2	UNIX overview		Q02 (Lecture 2)	
3	First Program		Q03 (Lecture 3)	A01-Poem
4	Escape Sequences, variable names		Q04(Lecture 4)	A02-Escape Characters
5	Primitive data types, relational operators		Q05(Lecture 5)	
6	Working with numbers		Q06(Lecture 6)	
7	Mixing numbers		Q07(Lecture 7)	A03-Tip Calculation
8	String methods		Q08(Lecture 8)	A04-Phrase
9	Strings and Random		Q09(Session 9)	A05-Dice
10	User Input			
11	Midterm 1 Review			
12	Exceptions			
13	What-if?	SBL1-code, inspection forms and SBL1 session video	Q10 (Lectures 12 and 13)	A06-BMI Calculation
14	Comparing			
15	Java on the Web		Q11 (Lecture 15)	
16	Loops		Q12 (Lecture 16)	A07-Drawing Applet
17	GUI Elements			
18	The Interactive GUI	SBL2-code, inspection forms and SBL2 session video	Q13 (Lecture 17 and 18)	A08-Game Applet
19	Midterm 2 Review			
20	Objects			
21	Exceptions – Creating your own		Q14 (Lecture 20 and 21)	A09-CD class
22	Arrays			A10-CD Exception
23	Static Methods			
24	ArrayLists, Wrapper classes			A11-Array of Grades
25	Files	SBL3-code, inspection forms and SBL3 session video		A12-Array of CDs
26	Final Exam Review	Modified MSLQ		

Figure 20. Data Collection Schedule

SBL-Online Course Elements Analyzed

During the Fall 2010, the ICS111 group experienced SBL-Online. During the first session there were four teams composed of three students each for a total of 12 students. Experience shows that at this point, most of the students that will drop the class have already done so. The second and third session had three teams each. The second session, counted with two teams of four students and one team of three students for a total of eleven students. During the last session, two out of the three teams were composed of three students and one team had four students for a total of ten students.

Students were assigned to teams by ranking them according to their grades. Students were unaware of this procedure. Rankings took place three times during the semester before each SBL-Online session. This allowed students to move from one team to another and to meet more of their peers. Students' roles were also rotated so that all of them had the opportunity to be authors as well as evaluators. Aside from student ranking, another factor influencing the structure of the teams was the rotation of the author role. An effort was made to provide all students with the experience to have their assignments critiqued.

For privacy reasons, students were assigned a codename. This codename was strategically built to allow for easy student tracking throughout the semester. Codenames were assigned as follows: A single letter of the student name or last name followed by three digits. These numbers represented the team they were part of during each one of the sessions. Since there were three sessions, there were three numbers following the initial letter. For example a student X231 denotes a student whose name or last-name contains the letter X who was in team 2 during the first SBL-Online session, on team 3 during the second SBL-Online session and on team 1 during the last SBL-Online session. Students that failed to participate in one or more sessions have zeroes in their codename, a zero indicating failure to participate in the given session.

The following is a list of the teams and their members. Codenames preceeded by an asterisk signal the authors of the code being critiqued. Codenames preceded by a pawn sign are the scribes, which are the students that were assigned the task of submitting the group inspection form and they are also responsible for setting the SBL-Online meeting day and time.

Session 1	Session 2	Session 3
Team 1	Team 1	Team 1
#B131	K411	B131
*C133	*M312	*K411
A100	#M212	#B321
	B213	
Team 2	Team 2	Team 2
*M212	P423	*M212
B213	#D323	#A432
#R230	*B321	M312
Team 3	Team 3	Team 3
#M312	*B131	*P423
*D323	#A432	#C133
B321	R230	D323
	C133	B213
Team 4		
#P423		
*A432		
K411		

Figure 21. SBL-Online Team Participants

Video Transcription Process

At the risk of losing objectivity, the decision was made for the researcher to transcribe the videos. It was deemed important to witness the way in which students behaved and interacted with each other as well as the exchange of technical knowledge within student-student interaction. The technical background and the experience teaching the course in question provided better understanding of student interaction. Being able to observe the SBL-Online recordings, helped to accurately summarize the session as well as to comment on student situations as the transcription was taking place. Such comments became the basis for session interaction highlights.

Video Transcript Format

The transcript was formatted chronologically to go from left to right and from top to bottom. The transcript is to be read the same way as English is read. If two utterances are in the same line, one in the chat and another in the talk, it means that they occurred at the same time. Transcripts originally had the following format:

Time	Chat	Microphone	Comments

Figure 22. The Initial Video Transcription Form

Timestamps were usually provided when topics changed. There is no real use for the timestamp other than to provide a fast way to refer to the original video in case there was a need for clarification. For this reason, timestamps have been omitted from transcripts in Appendix D.

The first attempt to code these videos was done independently by three coders. After a fruitless attempt, the coders reconvened and decided that it would be more efficient and accurate to code together. The process of coding independently still required meeting to discuss the work and to come to an agreement on code assignment. Coding together therefore will represent a unanimous decision for code assignment to every utterance throughout the process.

As the coding process began, it became evident that in many occasions, one code would not be descriptive enough for some utterances. The decision was made to add a secondary code for both the chat and the spoken communication. The primary code would qualify most part of what the student said, while the secondary code would complement the primary code and would be used only as-needed. These changes reflected in the transcription form, which acquired its final form as shown in Figure 23.

Time	Primary chat code	Secondary chat code	Chat	Primary talk code	Secondary talk code	Talk	Comments

Figure 23. The Final Video Transcription Form

The Process of Adapting a Code to Describe SBL-Online Interaction

Curtis and Lawson's in their article entitled "Exploring Collaborative Online Learning" (Curtis & Lawson, 2001) present a coding scheme used to describe utterances in online collaboration. This code was used as the basis to create the SBL-Online coding scheme. Some of the codes were borrowed exactly as they are. New descriptions and codes were developed to better categorize student interaction in an SBL setting and to directly address the research questions. Curtis and Lawson had organized their codes in five distinct categories. Two more categories were added for this study. Figure 24 shows the code adaptation used for this study. Curtis and Lawson's original code can be found on Appendix B.

Behavior categories	Codes	Description	Example
Group Process	GS	Group skills: a generic code applied to expressions that encourage group activity and cohesiveness.	If we stick together we can get it done on time
	OW	Organizing work: Planning group work; setting shared tasks and deadlines.	Who is going to fill out the form in Laulima? Ok, you are next.
	IA	Initiating activities: Setting up activities to get the work done	Lets start by listening to what the author has to say. Lets do it in the whiteboard.
	NTA	Notification Announcement: Notification of future or past events taking place during the session. Announcements.	Hi guys, I'm back we need to wrap up soon
	ME	Monitoring group effort: Comments about the group's processes and achievements. Comments about what is to be done immediately (real time)	do we all agree on this? Do you understand what I mean? Please wait and I will show you.
Contributing	HEG	Help giving: Responding to questions and requests from others. Prompted by HES.	To make it visible you should change the permissions to 644 you could use a variable to hold the value
	FBG	Feedback giving: Providing feedback on descriptions or explanations from others. Usually prompted by FBS, HYP or EKE.	When you do it like that it causes problems I think that is the best way to do it
	SKN	Sharing knowledge: Sharing existing knowledge and information with others. (not prompted) A comment/elaborated comment that may be given in addition to somebody else's HEG, FBG or SDR. May also be sharing the lack of knowledge	I found out that there is an easier way to do that…. In my case I decided to do blah.. blah.. first, The assignment says to do it by dividing the percentage. I don't know that. I have not idea.
	EXP	Explaining or elaborating: The author explaining the code, or anyone supporting one's own position (maybe following a challenge).	Here is my loop to verify the validity of the number. I did it in Here because otherwise my code would crash.
	CHQ	Challenging Questioning: Challenging the contributions of other members. Questioning procedures of other members.	I do not think that is the way to do it You are not declaring the variables at the top the way it should be done
	RID	Real-time, Instructing, Directing: step by step instruction/direction, while someone else performs the actions. (realtime)	Type the info in the textbox... then click the ok button. (real time)
Seeking Input	HES	Help seeking: Seeking assistance from others, specifically, from the ground up. Help on something that has not yet been implemented.	can anyone tell me how to do this?
	FBS	Feedback seeking: Seeking feedback for something that has been explained, shown or is of everyone's knowledge. Referring to something done or in progress.	what is your opinion on this? is it done correctly? Do you have any questions?
	ADE	Advocating effort: Urging others to contribute to the group effort.	Are you still with us?
Uncertain Assertions	HYP	Hypothesizing: Guessing the possible cause of something. Uncertain assertion, usually done while troubleshooting. Usually follows a EKE, FBS or HES. May be followed for another HYP, FBG, HEG or EXP	Maybe it works for you because you have an instance variable Perhaps you did not set the Unix permissions correctly I'm not sure if I should do A or B…
	EKE	Explaining known errors: from the author or from any student reporting his/her own mistakes without specifically asking for help, putting it out there…	that part is the messy part of the code. I did not know what to do
	COM	Comparing work	Her work is way clearer than mine Using switch made the code way shorter than mine
Reflection/ Monitoring	RM	Reflecting on medium: Comments about the effectiveness of the medium in supporting group activities.	This thing is slow! can you hear me? I couldn't hear a thing!
	RCW	Reflecting on coursework: Comments about the course process and their experiences. May also be reflection related to future activities.	Yes, I got marked off too. I watch the podcast sessions more than once. I always do my homework last minute I just need to pay more attention to the class and assignments.
	SDR	Sudden realization: Tthe realization of something that was unknown. Reflection on one's own actions. Maybe surprised.	Aha! No wonder it kept on crashing!!! If I had known it, I wouldn't have spent so much time trying to fix that.
	TLR	Think-out-loud Reflection: Reflecting own capacities, strengths and weakneses.	I can solve these problems in my sleep. I just cannot do that, it is hard! What did I do? I am confused
Social Interaction	SI	Social interaction: Conversation about social matters that are unrelated to the group task. This activity helps to 'break the ice'. - jokes	my family is flying in from Mexico. what Are you eating?
	EMS	Empathy-Sympathy: to be considerate, understanding, empathizing, apologizing	that is terrible. Been there… sorry, I didn't mean that
Grant Concede	ACK	Acknowledgment: to answer any of the group process or a yes/no question. Can be positive or negative.	ok, no, yes, thank you, roger that, lol, no way, :-), :-(
	AGR	Agreement: Supporting a view. It is un-called for and does not come from the one explaining or giving help/feedback but from someone else. It is a short answer that may include a short	I did it like that too. yep, it works well.

Figure 24. Transcription Code Used for This Study

Codes and Categories

Group process. This category is based on Curtis and Lawson's *planning* category (Curtis & Lawson, 2001, p. 148). Group process encompasses activities related to organizing and synchronizing group activity. They promote cohesiveness among group members.

GS–Group skills: a generic code applied to expressions that encourage group activity and cohesiveness.

OW–Organizing work: Planning group work; setting shared tasks and deadlines.

IA–Initiating activities: Setting up individual activities to get work done.

NTA–Notification Announcement: Statement of future, past or current events.

ME–Monitoring group effort: Comments about the group's processes and achievements. Comments about what needs to be done immediately (real time).

Contributing. This category is based on Curtis and Lawson's contributing category. Contributing incorporates student activities whose purpose is to contribute to the peer review process as well as to the knowledge and understanding of the course material.

HEG–Help giving: Responding to questions and requests from others. Prompted by HES (help seeking), but sometimes also by HYP (hypothesizing) or EKE (explaining known errors).

FBG–Feedback giving: Providing feedback on descriptions or explanations from others. Usually prompted by FBS (feedback seeking), HYP (hypothesizing) or EKE (explaining known errors).

SKN–Sharing knowledge: Sharing intelligence with others. This action is not prompted; it is spontaneous. Sharing knowledge consists of comments given in addition to somebody else's HEG (help giving), FBG (feedback giving) or SDR (sudden realization). This code also includes students sharing their lack of knowledge.

EXP–Explaining or elaborating: This code is mainly used for the author explaining his/her code. It may be given to others supporting their own position perhaps following a challenge (CHQ).

CHQ–Challenging Questioning: Challenging the contributions of other members. Questioning procedures of other members. It is an observation pointed out with disapproval. It may take the form of a demanding remark.

RID–Real time, Instructing, Directing: A student directing another in a step-by-step fashion to perform an activity. This action is always in real time.

Seeking Input. These codes are based on the Curtis and Lawson's codes with the same name. They are applicable when students seek assistance from others. Different codes are assigned depending on how the input is requested.

FBS–Feedback seeking: Students request input for something that has been explained, shown or is common knowledge. Feedback seeking can only be applied to an activity that is either in progress or has already been completed.

ADE–Advocating effort: An attempt to incorporate students that are not actively participating. It includes facts such as asking for an opinion or requesting someone to perform an action. ADE is not in real-time

Uncertain assertions. Uncertain assertions are not part of the Curtis and Lawson's code, they are specific to SBL-Online activities. They were created to encompass those instances where students express an educated guess about why a program behaves in a certain way. Predicting causes and effects are all part of uncertain assertions.

HYP–Hypothesizing: Presuming the possible cause of something. Hypothesizing usually takes place while troubleshooting. When students are facing with a problem, HYP comprises student's opinions on causes or effects. Hypothesizing can emerge from a variety of situations including EKE (explaining known errors), FBS (feedback seeking) or HES (help seeking). It may also complement FBG (feedback giving), HEG (help giving), EXP (explaining) or even another HYP (hypothesizing). Hypothesizing is always related to programming processes and structures.

EKE–Explaining Known Errors: This code is always from the perspective of a student reporting his/her own mistakes without specifically asking for help. It reflects the process of describing a known problem.

COM–Comparing work: Comparing two or more approaches to solving one problem. Comparing usually happens when an evaluator compares his code with the author's. Comparisons may however address course examples or comparisons among two or more student's approaches.

Reflection monitoring. This category is loosely based on Curtis and Lawson's group of codes with the same name. However, this version has an approach specifically designed to address the various form of reflection. It ranges from the vocalization of inner-thoughts to reflections on the task at hand. Reflective thoughts in regards to the communication media are part of this category as well.

RM–Reflecting on medium: Commenting about the effectiveness of the medium in supporting group activities. This includes, audio, video, bandwidth, hardware and software. Technology issues directly affecting student communication during the SBL-Online sessions are part of RM.

RCW–Reflecting on coursework: Comments about the course mechanics, forms of evaluation, or class rules. It may include the student's own class experiences. May also be reflection or advice related to future activities.

SDR–Sudden realization: Describes a surprising revelation. The unexpected awareness of something that was unknown. It may be a reflection of the person's own action or a reflection based on someone's utterance, demonstration or representation.

TLR–Think-out-loud Reflection: Reflecting own capacities, strengths and weaknesses. Denotes students' thinking-out-loud and verbalizing feelings about the process in question.

Social interaction. Social interaction was originally a single code in a category by itself. For the purpose of this study, social interaction has been partitioned into two separate codes, pure social interaction and interaction which denotes empathy.

SI–Social interaction: Exchange dialog concerning social matters that are unrelated to the group task at hand.

EMS–Empathy-Sympathy: Interaction that promotes or denotes sympathy or compassion. EMS applies when students are considerate, understanding, empathizing, or even apologizing.

Grant/Concede. This section is a new addition to the original code. Codes in this section are especially important in online interaction where participants need to monitor each other's understanding in order to make progress during SBL-Online sessions.

ACK–Acknowledgment: The short and precise answer to a question from anyone in the group. An acknowledgment can be positive or negative and it is usually simple like a "yes," "no" or even an "LOL." An acknowledgment must be prompted since it is voiced in response to a question.

AGR–Agreement: To concur, to support a view. An agreement is not prompted and does not come from the person explaining or giving help/feedback but from someone else. It is a voluntary contribution that may include a short comment.

Coding in Context

Coding was done depending on context. Codes are assigned not just for the utterance itself, but also for who produced it and under which circumstances. The scenario surrounding any verbal or written expression is as important for coding as the utterance itself.

The best example for coding in context is EXP (explaining). EXP is strongly tied to the scenario where the author explains his/her code to the peer evaluators at the beginning of an SBL-Online session. If any of the evaluators explained anything, even if related to the code in question, it would not be categorized as EXP. It would be deemed something else depending of the circumstances as well as the specific utterance.

The Chat and the Spoken Word

Chat dialog was delivered in small discrete and sometimes intertwined units.

Chat entities tend to be kept short. Sometimes long messages are broken into smaller entries. This represented a challenge because several consecutive entries shared the same code, becoming repetitive and increasing the counts.

Time	Primary Code	Secondary Code	Chat
29:53	SKN		P423: what I did, and this was probably a cumbersome way of doing it
	SKN		P423: was I set up two constant values, one that specified the point where a BMI is “low” and one where it is “high”
	SKN		P423: and then after calculating the BMI value I put in there if statements

Figure 25. Discrete Chat Pieces Sharing One Code (Session 1, Team 4, 29:53)

Spoken communication on the other hand is the opposite. It is delivered in long chunks that are for the most part not well articulated. Since spoken communication units were long, they became easier to categorize using both a primary and a secondary code.

During SBL-Online video transcription, spoken statements were separated where the speaker paused for one-second or more. If chat and spoken communication took place at the same time, an effort was done to match the timing when the chat appeared while someone spoke. In the case of two or more students chatting about different topics at the same time, there were no issues since the chat was transcribed as it appeared on the screen. However, when two or more people used the microphone at the same time, they garbled each other’s spoken words resulting in unintelligible noise.

Time	P	S	Chat	P	S	Talk
6:16	SI		K411: hey guys!			
	EMS		K411: sorry I'm late			
	SI		K411: had a final that ran late.			
	SI		B321: all g	FBS	IA	B321: hey K, Blanca told me to review your code. She opened your code, we were just going to review it without you, but, since you're here now, should we, I don't know, wait for her, and then have you open it or should we just keep reviewing it like this?
				FBS		B321: what do you think B131?
	FBG		K411: up to you guys	ME	FBG	B131: I think that K should just review the code with us, when Blanca comes up we can just tell her what happened. I think she'll be okay with it.
	FBG		K411: you can keep reviewing like this if you want	RID	FBS	B321: okay then, did the code compile?
	NTA		K411: and I'll just answer questions you guys have			
	NTA		K411: or point stuff out			

Figure 26. Example of Chat and Talk at the Same Time (Session 3, Team 1, 6:16)

Some instructions or long explanations given by the instructor during the SBL-Online were acknowledged, but not transcribed in full. Instructor contributions are out of the scope of this dissertation. Remarks deemed important for the purpose of this research are those amongst peers, especially those directed towards peer teaching and peer learning.

As mentioned before, the researcher transcribed the video seizing the opportunity to add comments during the transcription process. Comments contain observations made by the researcher during the transcription process. Comments became part of the session analysis and they were omitted from the transcripts in Appendix D since they are not officially part of the transcript.

Code Selection Process

Even when codes were designed to accurately represent interaction during SBL-Online sessions, it was difficult sometimes to decide which code to apply in a given situation. The following is a recollection of the most prominent issues faced during the coding process.

EXP vs. SKN (Explaining vs. Sharing Knowledge)

EXP (explaining) is an action only executed by the author in respect to the assignment being presented for critique. If at any time during the SBL-Online session someone presents her own assignment, then it shall be considered EXP as well, since the author role was passed to the person showing her own assignment.

Time	Primary	Sec	Chat	Primary	Sec	Talk
56:22:00				TLR		M312: I was just wondering.
	SI		B321: so D, any suggestions if I plan on continuing with CE?	SI		M312: Do you guys like wanna see my code? Because it closes if I put the wrong input.
				SI		M312: if not is all right. Doesn't really matter
	SI		D323: sure can you send it to us?			
	SI		B321: can D appoint you to a mod?			
56:43:00	SI		D323: oh maybe I can	COM		M312: oh, I think I'm in the mood now. So how would I open… tools, application sharing…[+] ok pico bmi.java so from the top
57:28:00	FBS		IM: Any questions?	EXP		M312: + import java.util.*; is because I used the new scanner, I mean, I used the scanner instead of the BufferedReader [+]
	SI		D323: got coffee?	EXP		M312: if you look at the code it looks super complicated but then, I wanted to do it this way so if I put any wrong input it just closes right away.
	SI		IM: >-)			
	FBG		D323: wow, a lot of nesting			

Figure 27. Temporarily Becoming an Author (Session 1, Team 3, 57:28:00)

If a non-author begins talking about his approach on the assignment being critiqued, it is not considered explaining because this person is not the author. It is either sharing knowledge (SKN), which is usually non-prompted, volunteered knowledge, or comparing (COM) if the person speaking is comparing the author's approach to solving a problem, against his own way of doing it.

OW, IA and ADE (Organizing Work, Initiating Activities and Advocating Effort)

Initiating activities (AI) is similar to organizing work but with regards to a single activity applicable to everyone in the group. OW, on the other hand, assigns different activities towards achieving a common goal. Advocating Effort (ADE), is about incorporating students that may be distracted or unresponsive. ADE is especially important in an online environment where students may be engaged in other activities besides the SBL-Online session.

FBG vs. HYP (Feedback Giving vs. Hypothesizing)

The following is an example extracted from the spoken area of session 1, team 3 around minute 19:48. The transcript explains a clear difference between FBG (feedback giving) and HYP (hypothesizing). Note that in FBG there is no hesitation from part of M312, whereas in HYP he uses the aphorism "I figure that" which denotes an opinion, a hypothesis.

Time	Primary	Sec	Chat	Primary	Sec	Talk
20:00	ACK		D323: That is true	FBG		M312: so, then if you put a number in there, and it doesn't fall in between 48 and 80, it will be else then "type in the correct number" or something like that.
	AGR		D323: Sorry trying to remember to do an else statement	RID		M312: else, you will need the open parenthesis after the else, and then, I mean, no, not parenthesis, that bracket thing, the open bracket and then I guess you can go down one line and then do a System.out.println of "please type in a number between 48 and 80 inches"…. Yes , or something like that. I don't have my code open up right now, but yeah.
	FBS		IM: Any questions? Or can I go away and come back in 10 more min?	FBS		M312: B, what do you think about that?
	FBG		B321: good for now	HYP		M312: cause I figure that the try only catches if the input is not a number.
	NTA		IM: ok, leaving.	ADE		B321: Yeah, that's true. And he has to capitalize the "System.out".

Figure 28. FBG vs. HYP (Session 1, Team 3, 20:00)

FBS vs. ME (Feedback Seeking vs. Monitoring Effort)

Feedback seeking (FBS) will always be prompted and never in regards to people's status. FBS is in regards to the activity taking place, it may be about the task or the SBL process. Monitoring Effort (ME), on the other hand will be about team members' status. ME is used to keep the team on task.

HES vs. FBS (Help Seeking vs. Feedback Seeking)

A student seeks help (HES) in situations where there is nothing to build upon. Help builds from the ground up. On the other hand, when a student seeks feedback (FBS) it is with regards to an artifact either partially or fully built. Feedback looks for an opinion that will either validate the work or suggest changes.

Video Analysis Findings

Video findings will first be described as narratives of each round of sessions followed by the individual sessions emphasizing event highlights, as well as assignment submission messages. To view full transcripts of SBL-Online sessions please refer to Appendix D.

Submission messages were not required from students but many of them wrote them. These messages greatly contribute to understanding the student's state of mind as they submitted and resubmitted their assignments. Selected submission assignments are shown here, however a full compilation of submission messages can be found in Appendix E.

Inside SBL-Online

During this study, ten SBL-Online sessions took place. The first round of SBL-Online sessions comprised of four sessions while the remaining two rounds had three sessions each. Students were required to submit the assignment for grading in order to be assigned into an SBL-Online team and participate in the SBL-Online sessions.

When students' assignments are graded, feedback was given as a personalized message followed by a table showcasing the assignment rubric with the student's points. Some students never learned where to locate assignment feedback. This was made evident by student comments in the end-of-semester questionnaire.

Disclaimer: Chat messages and assignment submission messages were cleaned in regards to spelling before adding them to this dissertation. However, an effort was made to keep as much of the original message.

First Round of SBL-Online Sessions

Error counts performed in order to set up this study show that students have problems validating user input. The first SBL assignment was given specifically to address exceptions and validations as seen in Figure 15. The assignment required students to write a Java program to calculate the body-mass index (BMI) as specified in assignment 6 listed in Appendix C. Feedback given to the students upon grading the original BMI program submission included remarks on the use of magic numbers, imports needed for exception handling, validations and comments on the fact that many times even when the user provided invalid data (for example if a person weight was 2 lb or the height 70 ft.), calculations still took place instead of providing an error message.

As evidenced by the session narratives, during the first round of SBL-Online sessions, students were unsure about the SBL process. Students were reminded that in preparation for the studio session, they had to review the author's assignment independently. During the studio session they were instructed to review it as a team.

Students spent most of the session trying to fix the author's code and exchanging ideas rather than filling up the inspection form. The purpose of SBL-Online studio sessions is to review work and to fill out the inspection form, not to complete or fix assignments. Filling the inspection form should be a guide for SBL-Online sessions, instead, it seemed as an afterthought in this first round of sessions.

The average length of the first round of sessions was 53.75 minutes. The longest session of the semester took place in the first round and was 77 minutes long.

Session 1 Team 2 included the lower-ranked students. During their SBL-Online session students in this team advocated most of the time to answer the inspection form questions. Their lack of knowledge may be the reason for this. It is easier to conduct the session addressing a list of questions and providing short precise comments than to fix the author's assignment or otherwise exchange knowledge.

Table 4. Session 1 Code Counts

Session 1		Team 1 - 1 hr 17 min						Team 2 - 34 min					
		Chat		Talk		Total		Chat		Talk		Total	
Categories	Codes	count	percent	count	percent	count	percent	count	percent	count	percent	count	percent
Group Process	GS	1	1.3%	1	0.5%	2	0.7%	0	0.0%	0	0.0%	0	0.0%
	OW	1	1.3%	5	2.5%	6	2.2%	1	0.5%	4	7.5%	5	1.8%
	IA	1	1.3%	3	1.5%	4	1.4%	3	1.4%	5	9.4%	8	2.9%
	NTA	9	12%	5	2.5%	14	5.1%	5	2%	8	15.1%	13	4.7%
	ME	0	0.0%	1	0.5%	1	0.4%	6	2.7%	2	3.8%	8	2.9%
Contributing	HEG	1	1.3%	3	1.5%	4	1.4%	6	2.7%	3	5.7%	9	3.3%
	FBG	0	0.0%	21	10%	21	7.6%	23	10.4%	1	2%	24	8.8%
	SKN	0	0.0%	7	3.5%	7	2.5%	3	1.4%	2	3.8%	5	1.8%
	EXP	4	5.3%	21	10.4%	25	9.1%	11	5.0%	0	0.0%	11	4.0%
	CHQ	0	0.0%	8	4.0%	8	2.9%	2	0.9%	0	0.0%	2	0.7%
	RID	0	0.0%	1	0.5%	1	0.4%	4	1.8%	5	9.4%	9	3.3%
Seeking Input	HES	1	1.3%	0	0.0%	1	0.4%	17	7.7%	0	0.0%	17	6.2%
	FBS	4	5.3%	27	13%	31	11%	15	6.8%	12	23%	27	10%
	ADE	1	1.3%	1	0.5%	2	0.7%	2	0.9%	0	0.0%	2	0.7%
Uncertain Assertions	HYP	0	0.0%	10	5.0%	10	3.6%	0	0.0%	0	0.0%	0	0.0%
	COM	0	0.0%	13	6.5%	13	4.7%	0	0.0%	0	0.0%	0	0.0%
	EKE	0	0.0%	2	1.0%	2	0.7%	0	0.0%	0	0.0%	0	0.0%
Reflection Monitoring	RM	7	9.3%	5	2.5%	12	4.3%	23	10.4%	6	11.3%	29	10.6%
	RCW	2	2.7%	15	7.5%	17	6.2%	12	5.4%	1	1.9%	13	4.7%
	SDR	1	1.3%	2	1.0%	3	1.1%	6	2.7%	0	0.0%	6	2.2%
	TLR	9	12%	26	13%	35	13%	11	5%	0	0%	11	4%
Social Interaction	SI	9	12%	5	2.5%	14	5.1%	27	12%	2	3.8%	29	10.6%
	EMS	2	2.7%	3	1.5%	5	1.8%	5	2.3%	1	1.9%	6	2.2%
Grant Concede	ACK	20	27%	6	3.0%	26	9.4%	35	16%	1	1.9%	36	13.1%
	AGR	2	2.7%	10	5.0%	12	4.3%	4	1.8%	0	0.0%	4	1.5%

Session 1		Team 3 - 1hr 4 min						Team 4 - 38 min					
		Chat		Talk		Total		Chat		Talk		Total	
Categories	Codes	count	percent	count	percent	count	percent	count	percent	count	percent	count	percent
Group Process	GS	0	0.0%	2	0.6%	2	0.4%	1	1.5%	1	1.1%	2	1.3%
	OW	0	0.0%	6	1.9%	6	1.2%	0	0.0%	5	5.4%	5	3.2%
	IA	5	2.7%	11	3.5%	16	3.2%	2	3.1%	4	4.3%	6	3.8%
	NTA	7	4%	7	2%	14	2.8%	4	6%	6	7%	10	6%
	ME	1	0.5%	24	7.6%	25	5.0%	2	3.1%	5	5.4%	7	4.5%
Contributing	HEG	1	0.5%	1	0.3%	2	0.4%	3	4.6%	4	4.3%	7	4.5%
	FBG	21	11.5%	50	15.9%	71	14.3%	5	7.7%	3	3.3%	8	5.1%
	SKN	0	0.0%	3	1.0%	3	0.6%	10	15.4%	6	6.5%	16	10.2%
	EXP	6	3.3%	8	2.5%	14	2.8%	0	0.0%	2	2.2%	2	1.3%
	CHQ	0	0.0%	2	0.6%	2	0.4%	0	0.0%	0	0.0%	0	0.0%
	RID	1	0.5%	21	6.7%	22	4.4%	1	1.5%	0	0.0%	1	0.6%
Seeking Input	HES	0	0.0%	2	0.6%	2	0.4%	0	0.0%	1	1.1%	1	0.6%
	FBS	18	9.9%	50	15.9%	68	14%	2	3.1%	11	12.0%	13	8.3%
	ADE	0	0.0%	4	1.3%	4	0.8%	0	0.0%	0	0.0%	0	0.0%
Uncertain Assertions	HYP	2	1.1%	10	3.2%	12	2.4%	0	0.0%	2	2.2%	2	1.3%
	COM	2	1.1%	0	0.0%	2	0.4%	0	0.0%	0	0.0%	0	0.0%
	EKE	0	0.0%	7	2.2%	7	1.4%	0	0.0%	5	5.4%	5	3.2%
Reflection Monitoring	RM	1	0.5%	2	0.6%	3	0.6%	6	9.2%	5	5.4%	11	7.0%
	RCW	1	0.5%	10	3.2%	11	2.2%	4	6.2%	5	5.4%	9	5.7%
	SDR	1	0.5%	4	1.3%	5	1.0%	0	0.0%	0	0.0%	0	0.0%
	TLR	7	4%	29	9%	36	7%	5	8%	6	7%	11	7%
Social Interaction	SI	82	45%	25	8%	107	21.6%	5	8%	2	2%	7	4%
	EMS	3	1.6%	2	0.6%	5	1.0%	4	6.2%	4	4.3%	8	5.1%
Grant Concede	ACK	22	12%	25	8%	47	9.5%	8	12%	4	4%	12	8%
	AGR	1	0.5%	9	2.9%	10	2.0%	3	4.6%	11	12.0%	14	8.9%

Session 1, Team 1 – Highlights

Participants in this session include A100 who took part only in the first SBL-Online session, and later dropped the class. B131 had both experience as a programmer in the 80s as well as experience working at a hospital. This is of particular importance because the assignment required them to write code to calculate the BMI (body-mass index) and B131's experience was highly relevant making him especially helpful to his classmates.

C133 is the author of the assignment being reviewed. She was aware of the errors in her assignment and she acknowledged it in the following submission message:

> *"Hey Ms Polo, I messaged you about my assignment on Skype, but I'm not sure if you got it. This is what it said:*
>
> *My program works except when I enter an invalid.. height, my program doesn't terminate. It still lets me put in a weight, and it calculates the bmi. with an incorrect weight, it just terminates.*
>
> *But I'd rather turn it in than not.*
>
> *So here ya go!"*

Reviewers compare what C133 did with their own assignment and offer their opinion about it. They commented about ineffective input validation and vague error messages. Other issues included the need for comments, the use of magic numbers and the inaccuracy of the selection statements.

Remarkable Moments:

B131 is trying to get C133 to figure out a mistake that he has already seen in her assignment. However B131 does not tell her what the problem is, instead he points at the problem line in the assignment and prompts her to figure it out.

B131: "don't run the code. Don't compile it. Just look at the two statements and see if you can make, just one simple change to make it work"

A100 : "oh yeah! I didn't catch that! That's a good catch!"

A100: I see what's going on, is that OR statement. So, it's true both ways. If it is equal to 25.

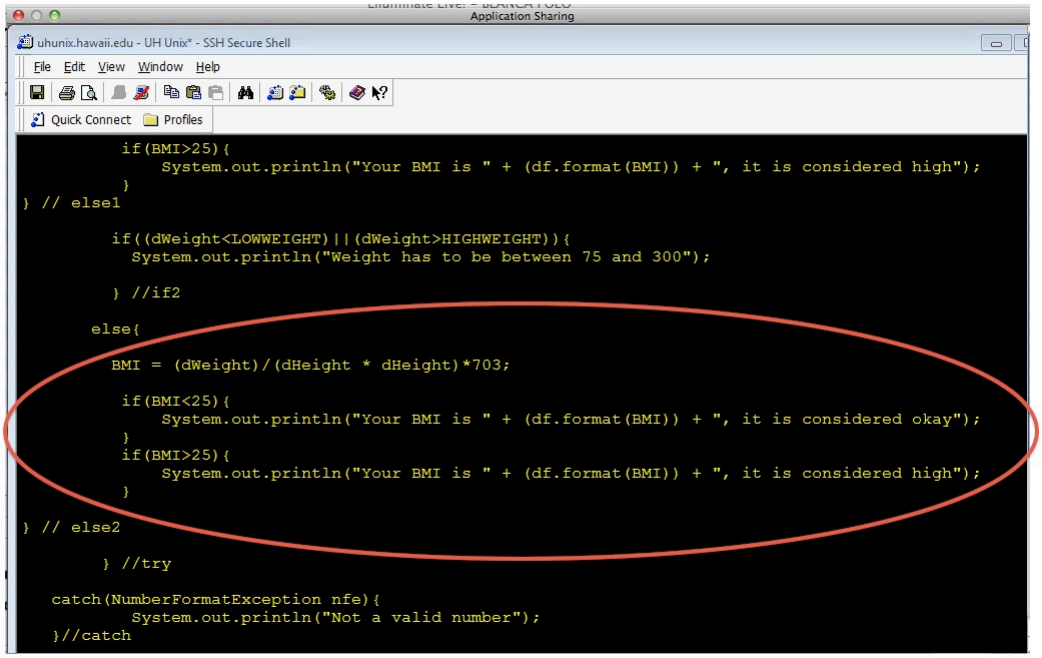


Figure 29. Error in the Code

During the resubmission, C133 wrote the following submission message:

This one is a LOT better. Or at least I think so :-)

C133's assignment resubmission was better designed and organized which resulted in improved performance.

Session 1, Team 2 – Highlights

This session is one of the shortest ones. It lasted only 35 min and 52 seconds, making it the shortest in the first SBL-Online round. Participants in this session include M212 who is the author, R230 and B213 as reviewers. B213 had previous programming

experience with Python, however it sees that it was detrimental to his learning in this class.

Interaction during this SBL-Online session is limited because students cannot talk due to audio issues. The author highlighted lines code and typed an explanation about them. He did this until he had gone through the whole assignment. After a brief and fruitless search for the assignment rubric they proceeded to fill out the inspection form in a very methodical way and without much thought.

The situation in this session was extreme in many ways. On one hand, the lack of audio prevented a richer and perhaps deeper communication. On the other hand, students in this session were ranked in the lower end of the spectrum. These facts may help explain the superficiality of their comments; Students mentioned the different selection tools that they used without any detailed information. They also commented that they all had points taken off because they used magic numbers.

Chat	**Talk**
B213: M, on line 24 what is the square looking box? M212: I have no idea M212: LOL B213: oh B213: LOL M212: she's using Pine, that's y M212: or I mean the SSH program B213: IC	B213 is pointing at the String[ ] args, precisely where the [and the] close together to form a square like this [].

Figure 30. Chat/Talk Interaction Discussion

Students in this group did not have enough exposure to the Unix environment. They referred to the square brackets "[]" as "the square." The square brackets are an integral part of a Java common instruction that goes at the top of most of the Java programs.

Submission Messages:

M212 (author): here is my assignment 6.My output is not correct I need help thank you.

R230: There isn't any need to check on the code for this one. I've been messing with it for a while and it still doesn't compile.

In regards to the original submissions, B213's assignment was incomplete and non-functional; R230's assignment didn't compile, had runtime errors and magic numbers; M212's assignment was the best of all and yet had validation problems.

B213 did not resubmit his assignment while M212 and R320 resubmitted a better version of theirs, however, none of them got full credit.

Session 1, Team 3 – Highlights

Participants are M312, D323 and B321. D323 is the author, while M312 is the scribe. Everyone in the team is able to hear. The author D323 does not have a microphone; therefore he has to explain his code using chat.

This team is reviewing the assignment and answering the inspection form at the same time. Participants are not following either the inspection form, or the assignment; the peer review is unorganized. In this session the seeking and the giving of feedback were done mostly using audio while D323 modified his assignment based on suggestions.

About 25 minutes into the session, and after moving around brackets on D323's code, they put the brackets back where they initially were. They compiled the code again, just to realize that their problem was actually a missing semi-colon. It was eye-opening to see that students weren't actually reading the error messages. They randomly modified the assignment in several occasions hypothesizing about what the problem could be, but nobody read the compiler's error message. Finally, the assignment compiled again 28 minutes into the session.

Submission Messages:

Based on these messages only one student, M312 felt confident that his assignment was working properly:

M312: tested in many different scenarios and they all worked.

D323 (author): Hi Blanca, here is the BMI code. just having trouble with logical sequence of where to place if's, else's, and try/catch. I suppose it will come with more practice.

B321: Prof. Blanca, I'm having a difficult time getting this code to work properly. My program compiles but I was unable to add the required "if" statements and my BMI comes out all wacky. Would I be able to go into the elluminate live session with my current .java file to learn more about what I am doing incorrectly? I'm hoping for any type of feedback as I'm sure it would help. I will keep trying to finish my code. Thank you.

During the session, D323 manages to explain his assignment via chat. In most of the SBL-Online sessions, EXP (explaining) accounts for a large percentage of the utterances, however, given the audio limitations, that is not the case in this session. As evidenced in Table 4, social interaction accounted for 21.57% of the utterances. This is the session with the most social interaction and 76.64% of it came directly from the chat.

B321 was expecting to learn from the scheduled Elluminate session. The assignment he resubmitted had some errors. M312 initially turned in a high-quality assignment but he didn't turn in the resubmission while D323, the author of the critiqued assignment was able to fix all the errors for resubmission.

Session 1, Team 4 – Highlights

Participants are P423, K411 and the author A432. During this session students touched on a variety of subjects that were related to the assignment at hand. They were comparing the "massive" nesting needed to validate as opposed to having a unique catch-

all statement allowing for a better organized assignment. They discussed the use of the System.exit(0) instruction to terminate execution as well as the importance of a well structured error message for the users.

The author A432 stressed the importance of looking at other people's code:

A432: ...for me I think I would have learned a lot more if I get to look at the 2 different ways that they did it and then just learned what I have learned so far it, and then I can kind-of combine it, will give me a better perspective, you know, get me more ideas on different ways in which I can improve my code. I know that we are doing this as a team effort too, I was just working on my code and find, that I think for me I can learn a lot more if I can see the other ways it was done too.

Submission Messages:

A432: I went around and around in circles on this one. I thought I had it going and then I started making changes to it and then sometimes I was on the right track, but many times it got worse. I know I'm missing some of the important components in the program. I just had some trouble trying to tie it all in together to make it work. At the least, I got it to be able to compile and run, but it is by no means, anywhere near complete.

K411: This assignment kicked my ass.

P423: ... I was unsure as to how to cause the program to close following invalid input from the user. If it was covered in sessions 12-13, maybe I'm just being a ditz and haven't found/recalled it...

In any case, I looked around online and found that the System.exit method can be used to terminate the program. If it was appropriate for us to use this method, then I guess I'm alright. If not, then I guess I'll find out the way it's supposed to be done soon!

Had fun making this one... despite the reminder of how unhealthy I really am >_>

This session was very productive. All members of this team successfully resubmitted their assignment for full credit. Many programming ideas were exchanged during this session. The most used code for this session was SKN (sharing knowledge), which is not prompted. When SKN follows FGB or HEG, it provides a more complete coverage of any topic.

Second Round of SBL: Online Sessions

During the second SBL-Online round of sessions there were a total of eleven assignment submissions. Students were distributed in two teams of four and one team of three. Table 5 shows the code distribution and category percentages for the second round of sessions.

Table 5. Session 2 Code Counts

Session 2		Team 1 - 40 min						Team 2 - 44 min						Team 2 - 40 min					
		Chat		Talk		Total		Chat		Talk		Total		Chat		Talk		Total	
Categories	Codes	count	percent	count	percent	count	percent	count	percent	count	percent	count	percent	count	percent	count	percent	count	percent
Group Process	GS	1	0.4%	2	1.4%	3	0.8%	0	0.0%	0	0.0%	0	0.0%	0	0.0%	0	0.0%	0	0.0%
	OW	0	0.0%	1	0.7%	1	0.3%	0	0.0%	2	1.1%	2	0.7%	0	0.0%	0	0.0%	0	0.0%
	IA	0	0.0%	2	1.4%	2	0.5%	2	1.9%	3	1.6%	5	1.7%	0	0.0%	1	1.8%	1	1.2%
	NTA	15	6.6%	3	2.2%	18	4.9%	11	10.4%	11	5.9%	22	7.5%	1	3.4%	0	0.0%	1	1.2%
	ME	3	1.3%	6	4.3%	9	2.5%	0	0.0%	0	0.0%	0	0.0%	1	3.4%	2	3.6%	3	3.6%
Contributing	HEG	0	0.0%	10	7.2%	10	2.7%	0	0.0%	11	5.9%	11	3.8%	0	0.0%	0	0.0%	0	0.0%
	FBG	36	15.9%	16	11.5%	52	14.2%	10	9.4%	31	16.6%	41	14.0%	7	24.1%	7	12.7%	14	16.7%
	SKN	16	7.1%	7	5.0%	23	6.3%	8	7.5%	10	5.3%	18	6.1%	2	6.9%	4	7.3%	6	7.1%
	EXP	2	0.9%	2	1.4%	4	1.1%	1	0.9%	5	2.7%	6	2.0%	0	0.0%	11	20.0%	11	13.1%
	CHQ	0	0.0%	0	0.0%	0	0.0%	0	0.0%	1	0.5%	1	0.3%	0	0.0%	1	1.8%	1	1.2%
	RID	11	4.9%	4	2.9%	15	4.1%	10	9.4%	5	2.7%	15	5.1%	0	0.0%	3	5.5%	3	3.6%
Seeking Input	HES	2	0.9%	2	1.4%	4	1.1%	0	0.0%	1	0.5%	1	0.3%	0	0.0%	1	1.8%	1	1.2%
	FBS	27	11.9%	30	21.6%	57	15.6%	20	18.9%	27	14.4%	47	16.0%	5	17.2%	11	20.0%	16	19.0%
	ADE	1	0.4%	3	2.2%	4	1.1%	0	0.0%	6	3.2%	6	2.0%	0	0.0%	0	0.0%	0	0.0%
Uncertain Assertions	HYP	3	1.3%	6	4.3%	9	2.5%	0	0.0%	1	0.5%	1	0.3%	0	0.0%	2	3.6%	2	2.4%
	EKE	0	0.0%	0	0.0%	0	0.0%	0	0.0%	1	0.5%	1	0.3%	0	0.0%	0	0.0%	0	0.0%
	COM	1	0.4%	0	0.0%	1	0.3%	0	0.0%	0	0.0%	0	0.0%	0	0.0%	0	0.0%	0	0.0%
Reflection Monitoring	RM	0	0.0%	1	0.7%	1	0.3%	4	3.8%	9	4.8%	13	4.4%	1	3.4%	6	10.9%	7	8.3%
	RCW	1	0.4%	6	4.3%	7	1.9%	1	0.9%	0	0.0%	1	0.3%	0	0.0%	0	0.0%	0	0.0%
	SDR	1	0.4%	2	1.4%	3	0.8%	0	0.0%	1	0.5%	1	0.3%	0	0.0%	0	0.0%	0	0.0%
	TLR	16	7.1%	8	5.8%	24	6.6%	1	0.9%	8	4.3%	9	3.1%	0	0.0%	4	7.3%	4	4.8%
Social Interaction	SI	40	17.7%	12	8.6%	52	14.2%	16	15.1%	30	16.0%	46	15.7%	3	10.3%	1	1.8%	4	4.8%
	EMS	3	1.3%	4	2.9%	7	1.9%	2	1.9%	5	2.7%	7	2.4%	1	3.4%	0	0.0%	1	1.2%
Grant Concede	ACK	44	19.5%	10	7.2%	54	14.8%	20	18.9%	14	7.5%	34	11.6%	8	27.6%	0	0.0%	8	9.5%
	AGR	3	1.3%	2	1.4%	5	1.4%	0	0.0%	5	2.7%	5	1.7%	0	0.0%	1	1.8%	1	1.2%

This time students were required to submit a message with their assignment. The message should contain the complete URL to their applet, which should be visible online.

There was some added complexity in this assignment given that students will have to deal with UNIX file permissions for online applet-posting.

In depth inspection of the second round of SBL-Online sessions illustrates the fact that less time is spent in fixing code and more time is spent in filling up the inspection form as well as giving and receiving feedback. The average session length shrank to 41.3 min. The longest of the sessions lasted 43min 49seconds.

In all instances of the second round, students organized themselves promptly as the session began in order to get everything that they need efficiently done. Session 2 participants spent an average of 57.89% of the time directly answering the questions in the inspection form. Students seem more confident, both presenting their code and giving feedback.

Common problems within this assignment include the application of random number generation for purposes beyond generating the number itself, action listeners, and posting the applet in a server for public access. These errors accurately match the expected errors as shown in Table 3.

Session 2, Team 1 – Highlights

Participants are K411, M212, B213, and the author M312. M212 is having a big problem because he does not understand how to post or view his applet online. His submission contains a local link to his assignment within his personal computer, instead of a URL.

Submission Messages:

> *M212: I still don't understand how to view my applet so I am pretty much shooting in the dark here.*

K411 and M312's submissions get full credit while M212 scored 80% of the points, despite the fact that he couldn't post his applet. B213's submission got only 23% of the credit, and he did not show up for the SBL-Online session.

This session is a clear example of how students empathize with each other and are willing to help. At minute 6:51 students discover that M212 cannot post or view his work online:

Time	Primary	Sec	Chat	Primary	Sec	Talk
	COM		M212: ur code looks nothing like mine			
	SI		M212: lol	FBS		M312: Really?! Did your code worked though?
7:38	FBG		M212: no I don't even understand how to view it			
	HES		M212: I need a tutor	ACK		M312: oh, ok!
	SI		M212: lol	HEG	EMS	M312: I would help you out if I'm not busy, I'm kind of busy, that's why. So... but I would help you out if I can.

Figure 31. Transcript Excerpt 1 (Session 2, Team 1, Minute 7)

Students spent 23.75% of total time of this session trying to help M212, which was not the author but was having serious problems. It is common for students not to ask the instructor for assistance when they are having problems; Students however, have no problem asking their peers for help.

Finally after almost 10 minutes of fruitless attempts to help M212, they move on but not without giving him hope:

> *M312: ... what we are going to do is lets try to answer all the questions, lets try finishing first and then we are going to help you out. So that we can get it over with the questions and the quizzes.*

M212 did not resubmit his code after the SBL session.

Session 2, Team 2: Analysis

Participants are P423, D323 and B321. B321 is the author and he is the only one in this team that didn't get full credit on this assignment. He wrongly manipulated a counter that ended up displaying a negative number of wins for a player. This was his only programming mistake.

Students showed more efficiency in this session, even when 15.7% of their communication was used for socializing. They spent 57.89% of the time answering

questions directly from the inspection form. They found humor as they helped each other as seen in the following transcript excerpt.

Time	Primary	Sec	Chat	Primary	Sec	Talk
	RM		P423: I'm hearing whirring sound in the background, it's hard for me to hear what you are saying	RM		D323: I'm sorry, let me switch down the [*]. Can you hear me now?
				RM		D323: I'm saying I need to open it up but the computer does not have that much memory so it lags pretty bad.
10:30	SI		B321: He's vacuuming at the same time	SI		D323: (laughter). Yeah Vacuum! Not good
	SI		P423: Ah	RID		D323: You wanna dive down and look at the…
				TLR		D323: Yeah, it looks logical
				TLR		D323: … it is separated
				SKN		D323:Just leaving a line between the east, the center and the west you can feel the difference. Just leave a line between the blank line
				SKN		D323:And between the east, the center and then west just for reading purposes.

Figure 32. Transcript Excerpt 2 (Session 2, Team 2, Minute 10)

Submission Messages:

P423: Once again thanks so much for your help with this!"

D323: So I finally got the applet to work after much frustration. I never expected to spend so much time devoted to a 5-color slot machine. Here it is:

P423 did not resubmit his assignment, while B321 and D323 resubmitted and got full resubmission credit.

Session 2, Team 3 – Highlights

During the grading of this assignment a mistake was made. A programming error was not identified by the instructor. The error, however was identified by a student during the individual code inspection. She presented it to the team during the SBL session. A432 ran the code dozens of times allowing her to identify an inconsistency in the results. During the SBL session, students asked the instructor questions related to this error. B131 was able to fix his error before resubmission. This is a reminder that not only the instructor can find errors in student work, other students also can. Providing a platform for this is a

characteristic of SBL-Online sessions. Regretfully, the audio for this session is extremely choppy and did not allow for the full transcription of the SBL-Online session.

A432: Even when it was matched up with the light blue color, sometimes it wouldn't count it as a won game. So I don't know… sometimes it counted it and sometimes not. I was trying to see if it was every other one but it was inconsistent… I don't know if anybody else noticed that …in your code?

After discussing that, the instructor showed up, asking if anybody had questions

B131: oh yeah, I have a question. When I generate a random number between 1 and 5, I'm supposed to get a number between 1 and 5. I shouldn't get a zero, correct?

IM: what happens is that if you have 5 as your limit, it will generate a number between 0 and 4 and then if you want to make it between 1 and 5, you have to add 1 so that it goes… because you know that by definition Java is 0-based.

A432: you defined your number range?

B131: ahm… I just picked 6 as my number range, so I thought I would be generating a number between 1 and 5. Maybe a zero is appearing over there. I have to double-check that.

Submission Messages:

A432: Hi Blanca, I emailed you earlier today on this.

R230: Well even with the extra day, I was unable to get this slot machine organized properly. As for posting this as a webpage, where is the reference for setting that up? I didn't even realize I had a web page.

C133: This is all I have so far. :(Every time I made a change, if my applet didn't change, I erased it so I at least had a starting point. I'm still trying to fix it though.

Resubmissions for this assignment were much better than the original submissions. R230 turned in his re-submission late, while the rest of the team got full resubmission credit.[1]

[1] Note: Full credit resubmission amounts to approximately 20% of the points that could have been earned with the initial submission.

Third Round of SBL-Online Sessions

The average length of the sessions decreased as students became more acquainted with the SBL-Online process. Sessions are longer in the first round but roughly the same in rounds two and three. Table 6 shows code percentages during the second round of SBL-Online sessions.

Table 6. Session 3 Code Counts

Session 3		Team 1 - 42 min						Team 2 - 44 min						Team 2 - 1 hr 5 min					
		Chat		Talk		Total		Chat		Talk		Total		Chat		Talk		Total	
Categories	Codes	count	percent	count	percent	count	percent	count	percent	count	percent	count	percent	count	percent	count	percent	count	percent
Group Process	GS	0	0.0%	0	0.0%	0	0.0%	2	3.8%	2	0.9%	4	1.4%	3	1.0%	0	0.0%	3	0.9%
	OW	1	0.9%	4	4.8%	5	2.5%	2	3.8%	1	0.4%	3	1.1%	0	0.0%	1	10.0%	1	0.3%
	IA	0	0.0%	2	2.4%	2	1.0%	1	1.9%	3	1.3%	4	1.4%	2	0.6%	0	0.0%	2	0.6%
	NTA	11	9.6%	4	4.8%	15	7.6%	4	7.5%	24	10.5%	28	10.0%	35	11.1%	4	40.0%	39	12.0%
	ME	2	1.8%	2	2.4%	4	2.0%	0	0.0%	2	0.9%	2	0.7%	0	0.0%	0	0.0%	0	0.0%
Contributing	HEG	2	1.8%	2	2.4%	4	2.0%	0	0.0%	0	0.0%	0	0.0%	13	4.1%	2	20.0%	15	4.6%
	FBG	15	13.2%	8	9.6%	23	11.7%	7	13.2%	41	18.0%	48	17.1%	30	9.6%	0	0.0%	30	9.3%
	SKN	5	4.4%	1	1.2%	6	3.0%	5	9.4%	21	9.2%	26	9.3%	15	4.8%	0	0.0%	15	4.6%
	EXP	10	8.8%	3	3.6%	13	6.6%	0	0.0%	0	0.0%	0	0.0%	27	8.6%	0	0.0%	27	8.3%
	CHQ	0	0.0%	3	3.6%	3	1.5%	0	0.0%	0	0.0%	0	0.0%	2	0.6%	0	0.0%	2	0.6%
	RID	2	1.8%	7	8.4%	9	4.6%	3	5.7%	17	7.5%	20	7.1%	16	5.1%	0	0.0%	16	4.9%
Seeking Input	HES	0	0.0%	1	1.2%	1	0.5%	0	0.0%	0	0.0%	0	0.0%	5	1.6%	0	0.0%	5	1.5%
	FBS	15	13.2%	16	19.3%	31	15.7%	7	13.2%	41	18.0%	48	17.1%	34	10.8%	3	30.0%	37	11.4%
	ADE	0	0.0%	3	3.6%	3	1.5%	0	0.0%	2	0.9%	2	0.7%	3	1.0%	0	0.0%	3	0.9%
Uncertain Assertions	HYP	2	1.8%	1	1.2%	3	1.5%	2	3.8%	3	1.3%	5	1.8%	3	1.0%	0	0.0%	3	0.9%
	EKE	2	1.8%	0	0.0%	2	1.0%	0	0.0%	0	0.0%	0	0.0%	5	1.6%	0	0.0%	5	1.5%
	COM	0	0.0%	0	0.0%	0	0.0%	0	0.0%	1	0.4%	1	0.4%	0	0.0%	0	0.0%	0	0.0%
Reflection Monitoring	RM	0	0.0%	1	1.2%	1	0.5%	1	1.9%	7	3.1%	8	2.8%	14	4.5%	0	0.0%	14	4.3%
	RCW	1	0.9%	1	1.2%	2	1.0%	0	0.0%	8	3.5%	8	2.8%	4	1.3%	0	0.0%	4	1.2%
	SDR	1	0.9%	0	0.0%	1	0.5%	0	0.0%	3	1.3%	3	1.1%	2	0.6%	0	0.0%	2	0.6%
	TLR	3	2.6%	5	6.0%	8	4.1%	0	0.0%	21	9.2%	21	7.5%	11	3.5%	0	0.0%	11	3.4%
Social Interaction	SI	17	14.9%	2	2.4%	19	9.6%	10	18.9%	1	0.4%	11	3.9%	22	7.0%	0	0.0%	22	6.8%
	EMS	6	5.3%	0	0.0%	6	3.0%	3	5.7%	1	0.4%	4	1.4%	10	3.2%	0	0.0%	10	3.1%
Grant Concede	ACK	18	15.8%	14	16.9%	32	16.2%	3	5.7%	23	10.1%	26	9.3%	47	15.0%	0	0.0%	47	14.5%
	AGR	1	0.9%	3	3.6%	4	2.0%	3	5.7%	6	2.6%	9	3.2%	11	3.5%	0	0.0%	11	3.4%

For this session most of the errors were product of incorrect validation and flawed looping. In both cases, students have problems with creating the right conditions to adequately control the flow of execution of their code. Many students demonstrated that they had the right idea to solve the problem. However, they did not know how to do it. This problem is "Background Error B3," as noted in Figure 4. It still remains unclear how to help students overcome this impasse. SBL may provide help, since all resubmissions from non-full-credit students show improvement.

Session 3, Team 1 – Highlights

Participants were B131, B321 and the author K411. Extra time was spent on logistics because the author, K411, arrived late and it is usually the author who explains the code at the beginning of the SBL-Online session. This roughly doubled the set up time because, students had first set up to work without K411 and later when he joined they had to reorganize themselves. Despite the eventualities students effectively reviewed the code, and their interaction allowed them to turn in improved resubmissions.

Time	Primary	Sec	Chat	Primary	Secondary	Talk
6:16	SI		K411: hey guys!			
	EMS		K411: sorry I'm late			
	SI		K411: had a final that ran late.			
	SI		B321: all g	FBS	IA	B321: hey K, Blanca told me to review your code. She opened your code, we were just going to review it without you, but, since you're here now, should we, I don't know, wait for her, and then have you open it or should we just keep reviewing it like this?
				FBS		B321: what do you think B131?
	FBG		K411: up to you guys	ME	FBG	B131: I think that K should just review the code with us, when Blanca comes up we can just tell her what happened. I think she'll be okay with it.
	FBG		K411: you can keep reviewing like this if you want	RID	FBS	B321: okay then, did the code compile?
	NTA		K411: and I'll just answer questions you guys have			
	NTA		K411: or point stuff out			

Figure 33. Transcript Excerpt 3 (Session 3, Team 1, Minute 6)

Submission Messages:

Submissions from K411 and B321 received similar feedback, emphasizing that even when they did not complete the requirements, they had the right idea about the process to finish the assignment. B131 had full credit. This time K411 added a submission message:

K411: This assignment was no fun. But it's done! Yay!

Resubmissions showed great improvement over the original submissions. All of them got full credit.

Session 3, Team 2: Analysis

Participants are M312, A432 and the author M212 is absent. A432 and M312 commented on the difficulties they would have to face in order to reschedule the SBL-Online session so close to the end of the semester. It was suggested by the instructor, that they evaluated the author's work without him being present.

The author's code was complete and with more functionality than that of the reviewers. In absence of the author reviewers assigned the code extremely low scores during the group inspection. Scores were even lower than scores originally assigned during the individual code inspection. When compared to the rest of the sessions this review was the lowest one. This situation underscores the fact that if the author is missing there is nothing preventing reviewers from giving harsh comments.

This is a set of rough comments in regards to the absent author's code:

M312: if I look at the variables, I can see, why he named them that way so like, S1obj, I think it means string 1 object. But that has really nothing to do with that variable. So I would say one or two?

M312: and then, number 8. For the following statements, no no, the comments and documentation where clear and helpful. I don't see comments. Or documentation.

A432: yeah, there wasn't really any.

M312: I'm thinking one or two

A432: I guess one, because, I will say one because there wasn't really anything that could have helped me.

M312: I was thinking of two because of the errors that he gave. Earlier, but one is yeah, just one.

M312: All right, so number 9, the purpose of the program was clear. No, just looking at this so like a quick glance is not telling me anything because no comments or anything like that if I look through it and I spend like quite a while on it then I could probably figure it out but what would you think of that? The purpose of the program was clear.

A432: I don't know, I say one or two

M312: okay yeah you think it was good enough? Because I think if we spend a little time looking at it we could figure it out. Is not something but I would argue about.

A432: two in that.

M312: yeah, I'm not so happy with that so I will put one.

Submission messages (M212 was the absent author):

M212: here is my assignment 12. My output is not exactly how you wanted it to be.

A432: Sorry... I really messed up on this assignment.

M312: After working on it for 7 hours, I couldn't think straight anymore. I hope I'll learn something new from Elluminate session. Keep getting mismatch or incompatible error. I think I understand the error but I don't know how to fix it.

While M212 was not present and he didn’t resubmit, here are the resubmission messages of the reviewers:

> *A432: "I feel so much better after working on this resubmit and seeing how I was able to work out a lot of the issues I had with my original submission. It looks so much better. I just got really discouraged and wasn't too motivated to complete it because I was having so much trouble on it. But I am glad I redid it.*
>
> *Anyway, thank you for putting up with me. This was a really tough semester for me with 5 classes, along with having so much difficulty with my other computer science class (discrete mathematics). I really learned a lot in your class and I had some fun in the process."*

> *M321: resubmit. I made a huge improvement but still failed at it. I don't know if I should still turn it in but I'll do it anyway.*

M212’s assignment was noticeably better than the reviewers’, which should call for different comments not the ones given. M212’s code made more points in the submission without resubmission than the points that the others made adding submission and resubmission.

Session 3, Team 3 – Highlights

Participants were C133, D323, the author P423 and a late addition to the team B213. When the session begins one of the team members, C133, is missing. However, with the late addition of B213 to the team there was be no shortage of peer evaluators. This session runs entirely on chat. D323 is in the library, while P423 and B213 have audio problems. This session was held entirely via chat and ran for over an hour.

Evaluators found duplicate code within the assignment. The session revolves around trying to delete any redundant instructions. By the end of the session P423 manages to remove redundant code successfully. Students in this team find Elluminate (SBL-Online sessions) quite helpful. For full session transcripts please refer to Appendix D.

time	primary	secondary	chat
	EXP		P423: so I tried taking out this highlighted section entirely
	FBG		D323: looks like you repeat code
	EXP		P423: in the CDshelf class
	EXP		P423: but when I did that, and then tried to leave a blank artist/album name when running the code
	EKE		P423: the program crashed
	ACK		B213: oh
	FBS		B213: was there something missing?
	NTA		B213: I'm trying to look it over now
	EXP		P423: I'm not sure :/ I just know that with this highlighted bit left in the program worked.
18:30	NTA		D323: MMMM, trying to figure it out
	RCW		P423: so after a while I had to give up and submit this as is >_>

Figure 34. Transcript Excerpt 4 (Session 3, Team 3, Minute 18)

Submission Messages:

*P423: I tried to make changes to the CD class, but all I ended up doing was breaking the program, and again, I think it at least already works fine as it is (I'm pretty sure that it won't create any invalid CD objects). So just settling for this for now. Might be able to make some changes to it after the elluminate session ^^*

C133: Once again, I'm completely lost. Are we reading the user's input, and then printing it out like your driver classes did? And then, I think I'm lost as to how I link up my array list with my CD object. (Besides creating a program like your previous driver classes. I'm gonna keep working on this until I figure it out, but I'd rather turn it in now. Again (not a lot to show for it), but that's because I have dummy programs with a whole bunch of random code on them!

D323: I just realized assignment #12 is due 6hrs before #11 , so I honestly never got a lot done. I understand what needs to be done and how to do the logic but just not enough time, sorry.

Resubmission Messages:

P423: The program itself doesn't seem to run much differently than before, if at all, but the code seems a lot more "balanced" now thanks to the elluminate session. It was very helpful :)

C133: FINALLY! Thank goodness for Elluminate! Thank you for all of your assistance this semester. Arrays are still gonna take a lot of work for me to fully understand, but hopefully I can get a B this semester :) so I can take 211 and 215 next semester.

B213 and D323 did not resubmit their assignments.

Sessions Overview

This section will begin by explaining the process of maturation that students underwent as they became more familiar with the mechanics of the SBL-Online. It will continue with the in-depth analysis of video coding and how it addresses the research questions.

Becoming Efficient

As students participated in more SBL-Online sessions, they became more experienced with the process while becoming more efficient in producing the necessary deliverables. During the first round of sessions, students were unfamiliar with the process. They took more time trying to fix the author's code than evaluating it and filling out the group inspection form. After the first round of sessions, students spent less time trying to fix the code and more time working on the SBL-Online deliverables. A sign of

efficiency emerged when students decide to abide by form using it as a guide during the SBL-Online session. An example is featured in Figure 35.

Time	Primary	Sec	Chat
21:43	RID		K411: so, M. I'll just throw up the questions and you can type your answer, 1 to 5
	ACK		M212: ok
	NTA		K411: I 'll average it with mine and enter that
21:55	FBS	RID	K411: So, easy to read?
	FBG		M212: 5
22:06	FBS	RID	K411: variable names were descriptive and relevant?
	FBG		M212: 5
22:20	FBS	RID	K411: properly indented
	FBG		M212: 5

Figure 35. Becoming Efficient (Session 2, Team 1, 21:43)

Code Counts' Analysis

Just like students became more efficient, so did the video coders. The more coding that was done, the faster it was to unanimously select the codes to be assigned. Similar actions resulted in similar coding throughout the inspection form-filling process. For example, FBS RID was adopted for each question read by a team member. Reading a question from the form was categorized as seeking feedback combined with real time instructing to get the combined answer into the form. Therefore FBS/RID + FBG became a pattern of efficient form filling. Based on patterns like this, in code counts, and the thick data collected, the session transcripts will be analyzed.

Is there any indication that learning is taking place?

In order to identify learning there should be a discussion followed by an agreement in regards to something. Using code patterns to visualize the process may not be accurate because codes are assigned in context. Another problem is that conversations in the chat tend to intertwine and can potentially obliterate any plausible pattern.

Even with these limitations in mind an effort was made to quantify learning moments by identifying patterns as listed in the following equation:

Learning Moment =
Any one of the following codes, FBS|HES|ADE|EKE|COM|SDR|TLR]
Followed by any one of these codes [HEG|FBG|SKN]

Table 7. Learning Moments Counts

Session	Learning Moment Pattern Count
Session1-Team1	14
Session1-Team2	17
Session1-Team3	27
Session1-Team4	6
Session2-Team1	19
Session2-Team2	23
Session2-Team3	5
Session3-Team1	13
Session3-Team2	30
Session3-Team3	16

A Perl script[2] was used to find and count these patterns within the session transcript files, and the information in Table 7 was revealed. However, given the coding limitations listed above, it is possible that the most accurate way to find out if there is any potential learning within a session can be found in the counts of those codes used in "contributions" such as HEG (help giving) and FBG (feedback giving) plus CHQ (challenging questioning) and RID (real-time instructing directing). CHQ may be included because challenging a fact provides a learning opportunity. RID may be included because students direct each other step-by-step on how to accomplish a task.

HEG and FBG are de-facto contributions to somebody's learning. Helping someone usually is done from the ground-up. Giving feedback results from the evaluation of an

[2] Script can be found in Appendix D.

idea or artifact. Both codes are usually prompted. However there is not a one-to-one relationship between FBS and FBG or HES and HEG as seen on Table 8.

Table 8. Code Counts Sorted by Count per Session

Session 1								Session 2						Session 3					
Team1		Team2		Team3		Team4		Team1		Team2		Team3		Team1		Team2		Team3	
Code	Total	Code	Total	Code	Total	Code	Total	Code	Total	Code	Total	Code	Total	Code	Total	Code	Total	Code	Total
TLR	35	ACK	36	SI	107	SKN	16	FBS	57	FBS	47	FBS	16	ACK	32	FBG	48	ACK	47
FBS	31	RM	29	FBG	71	AGR	14	ACK	54	SI	46	FBG	14	FBS	31	FBS	48	NTA	39
ACK	26	SI	29	FBS	68	FBS	13	FBG	52	FBG	41	EXP	11	FBG	23	NTA	28	FBS	37
EXP	25	FBS	27	ACK	47	ACK	12	SI	52	ACK	34	ACK	8	SI	19	SKN	26	FBG	30
FBG	21	FBG	24	TLR	36	RM	11	TLR	24	NTA	22	RM	7	NTA	15	ACK	26	EXP	27
RCW	17	HES	17	ME	25	TLR	11	SKN	23	SKN	18	SKN	6	EXP	13	TLR	21	SI	22
NTA	14	NTA	13	RID	22	NTA	10	NTA	18	RID	15	TLR	4	RID	9	RID	20	RID	16
SI	14	RCW	13	IA	16	RCW	9	RID	15	RM	13	SI	4	TLR	8	SI	11	HEG	15
COM	13	EXP	11	NTA	14	FBG	8	HEG	10	HEG	11	ME	3	SKN	6	AGR	9	SKN	15
RM	12	TLR	11	EXP	14	EMS	8	ME	9	TLR	9	RID	3	EMS	6	RM	8	RM	14
AGR	12	HEG	9	HYP	12	ME	7	HYP	9	EMS	7	HYP	2	OW	5	RCW	8	TLR	11
HYP	10	RID	9	RCW	11	HEG	7	RCW	7	EXP	6	IA	1	ME	4	HYP	5	AGR	11
CHQ	8	IA	8	AGR	10	SI	7	EMS	7	ADE	6	NTA	1	HEG	4	GS	4	EMS	10
SKN	7	ME	8	COM	7	IA	6	AGR	5	IA	5	CHQ	1	AGR	4	IA	4	HES	5
OW	6	SDR	6	OW	6	OW	5	EXP	4	AGR	5	HES	1	CHQ	3	EMS	4	EKE	5
EMS	5	EMS	6	SDR	5	COM	5	HES	4	OW	2	EMS	1	ADE	3	OW	3	RCW	4
IA	4	OW	5	EMS	5	GS	2	ADE	4	CHQ	1	AGR	1	HYP	3	SDR	3	GS	3
HEG	4	SKN	5	ADE	4	EXP	2	GS	3	HES	1	GS	0	IA	2	ME	2	ADE	3
SDR	3	AGR	4	SKN	3	HYP	2	SDR	3	HYP	1	OW	0	EKE	2	ADE	2	HYP	3
GS	2	CHQ	2	RM	3	RID	1	IA	2	EKE	1	HEG	0	RCW	2	COM	1	IA	2
ADE	2	ADE	2	GS	2	HES	1	OW	1	RCW	1	ADE	0	HES	1	HEG	0	CHQ	2
EKE	2	GS	0	HEG	2	CHQ	0	COM	1	SDR	1	EKE	0	RM	1	EXP	0	SDR	2
ME	1	HYP	0	CHQ	2	ADE	0	RM	1	GS	0	COM	0	SDR	1	CHQ	0	OW	1
RID	1	EKE	0	HES	2	EKE	0	CHQ	0	ME	0	RCW	0	GS	0	HES	0	ME	0
HES	1	COM	0	EKE	2	SDR	0	EKE	0	COM	0	SDR	0	COM	0	EKE	0	COM	0

In all but one of the sessions, the amount of HES (help seeking) is significantly lower than the tallies of HEG (help giving). This shows that directly asking for help is uncommon but offering help is widespread. Help giving (HEG) can be prompted directly, or indirectly.

FBS (feedback seeking) happens slightly less often than FBG. This fact shows that feedback is often sought. This coincides with the purpose behind every SBL-Online session, which is to evaluate and to provide feedback. It also shows that students are not afraid to ask their peers for feedback.

In most instances the counts of EXP (explaining) done by the author and SKN (sharing knowledge) were similar. While the author must explain the code at the beginning of the SBL session, sharing knowledge is contributing to the explanation and elaborating on contributions by others. As evidenced by Table 9, contributing codes make up 28.65% of the total count. There are notable exceptions in which EXP is very low, the

following are examples of this: In S1-T4 the author seems shy and instead of explaining her code she hypothesizes, reflects and asks for feedback. In S2-T1 the author got full credit in his assignment and he gave only four very short straightforward explanations, while big part of the session was spent trying to help another student. The most extreme of these situations was the session in which the author was not present. That one had no EXP. This information can be seen in Table 8.

Table 9. Code Categories Percentages

Categories	Session1	Session2	Session3
Group Process	12.80%	9.03%	13.97%
Contributing	22.78%	31.13%	32.04%
Seeking Input	13.97%	18.33%	16.21%
Uncertain Assertions	4.41%	1.89%	2.37%
Reflection Monitoring	17.62%	9.43%	10.35%
Social Interaction	15.05%	15.77%	8.98%
Grant Concede	13.38%	14.42%	16.08%

CHQ (challenging questioning) is also related to learning. CHQ prompts learning opportunities since those being challenged have to defend their points of view. On the other hand codes like RID may not necessarily represent a learning opportunity. RID was used in cases where one student is directing another in a step-by-step fashion pointing what to do while the other one follows, perhaps automatically.

Overall, the code categories presented on Table 9 show that the contributing category dominates every session while "uncertain assertions" display the lowest percentage in all instances as well. Adding the percentages of "seeking input" and "uncertain assertions" does not add to "contributing." This shows that students contribute information voluntarily, without being prompted.

After reviewing student-student interaction within the treatment group, it was necessary to look into the interaction that had taken place within the control group. To achieve this, the approach was completely different since there were no recorded sessions. It was necessary to resort to the thick data collection.

Student-Student Interaction During the Traditional Implementation

The following information is the result of email messages collected during the non-SBL semester. In this example, there is evidence of traditional, non-SBL students' perspectives in regards to meeting other students and interacting with them.

H and R had become friends during the semester. They were both taking the online ICS111 class, as well as a face-to-face UH course. R had some programming experience and she was a declared computer science major. H was an engineering major just trying to get Java under her belt. Here are some email comments from both R and H

> *"R really helped me out. She showed me how use a program called eclipse which makes it easier to see your errors as well as runs the program all in the same place. Very user friendly. Once I got it to where I wanted it I moved it to Unix. The stuff I was having trouble on turned out to be a lot easier than I thought. I was overthinking it. Now to study for the midterm. Here is my link to my Slot Machine. I hope you like it."*

In another instance, they requested an Elluminate session specifically to ask questions about an assignment. These are the comments after the online meeting with the instructor:

> *"You're probably thinking the same thing as us - FINALLY!!!*
>
> *I'm not sure how I did with this one because it was not easy to figure out how to make it work.*
>
> *Well, I'm moving on to try to study for the final. I've got a few days to cram…"*
>
> *- - - - -*
>
> *"Thank you SO much, Blanca for coming online late at night to help us. We were really trying to work it out only our logic was a little mixed up.*
>
> *We actually knew where the zero was coming from but we though we could change it to -1 after the program. It "might" have eventually come to us later, but our brains were getting a little fried.*
>
> *I think that it works now that you gave us all those hints."*

As mentioned before, assignment submission messages are not required, however, students tend to send them accompanying their submissions. Submission messages from the control group are radically different to those from the SBL-Online group. Appendix E contains all the assignment submission messages.

Messages like the one below will be referred as "terminal." These messages, in many instances denote hopelessness. They are written by students turning in their assignments just to avoid getting a zero. Doing this leaves a void in students' knowledge, making it harder to successfully execute subsequent assignments. To showcase this situation, here are some sample terminal submission messages from the control group:

Student "terminal" messages:

"I'm not completely happy with what I'm turning in..It works nice..up to a point. I just wanted to turn in something just in case I cant figure out what I did wrong."

"I am sending you my assignment 6 BMI code. I know that it compiles but I know that it doesn't work exactly correct. I am sorry that I didn't have the time to put into this but I have had a very hectic week and other extenuating circumstances, (not to make excuses). I hope you will take this into consideration while grading my assignment. Thank you for your time and have a good weekend."

Here are messages from students that voluntarily submitted, fixed and resubmitted their assignment:

Submission:

"Here is my slotmachine code. I had trouble getting the actionlistener to work when the code was with it, so I moved it up and it worked. You can find the slotmachine at the same address as before"

Resubmission: - - -

"I hade some additional thoughts and made some changes to my program. I feel that it makes it look cleaner, I hope you like it. The address is ___."

- - - - - -

Submission:

"My slot machine is incomplete and not working. I finally managed to get it to show up, but now I gotta head to work and I don't think I'll be able to finish it in time so I'll just submit what I can. If I can get it finished I'll submit it ASAP."

Resubmission: - - -

"Second version. Did what I could, I got the counter working but no random number generation or colors changing. This is what I could do before the deadline, but I will finish it up tomorrow morning."

Specific individualized feedback was provided to students if they emailed the instructor their questions before the deadline. Students knew that they could always ask for help. After submission, feedback was provided to all students, however, not all of them received it. In order for feedback to work it must be received, assimilated and put back into practice. There is no way to verify if this happened. Students may have ignored the feedback leaving errors unaddressed.

There is another kind of message that will be labeled as "non-terminal." Messages asking the instructor for feedback after submitting the assignment for grading are non-terminal. These messages denote student willingness to continue working on mastering the skills required by the assignment. The following is an example of a submission, non-terminal Message:

"This assignment really gave me a tough time. I could not determine how to prevent the user from entering no numeric characters, or just pressing enter while using the scanner class. I ended up using the buffered reader. Can you give me some ideas on the scanner problem? Have a nice weekend."

Feedback is given to all students, nevertheless, some students ask specifically for feedback upon assignment submission. On the other hand, there are those who did not

read or address the feedback received. At the end of the course, evidence found suggested that some students didn't know where assignment feedback was located. Evidence of this was found in the end-of-semester questionnaire.

The following is a submission message from a traditional student that would have benefitted from participating in SBL sessions:

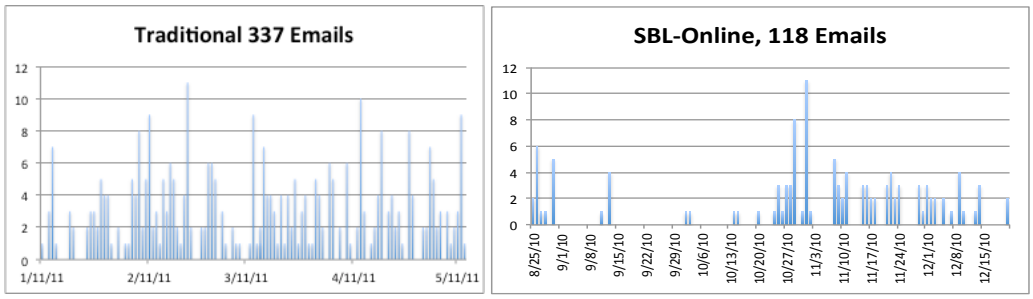
> *"Whew! What a challenge this was… It made me wonder what other students' outputs look like. I'd be curious to see what they've come up with since there are probably lots of differences..."*

Comparing SBL-Online vs. Traditional Implementations

The thick data collection afforded by the use of the Laulima course management system as well as the UH email system provided this study with multiple perspectives for course comparison.

Student emails were collected during both the traditional and the SBL-Online implementations of the same course. During the traditional implementation, 337 emails from students were received. During the SBL implementation only 118 emails were received. Most of these emails contained questions in regards to the assignments, course policies, practices or grading.

Table 10. Emails Sent to the Instructor



During the traditional implementation, students emailed the instructor for help and guidance. During the semester with SBL students found the answers to similar questions

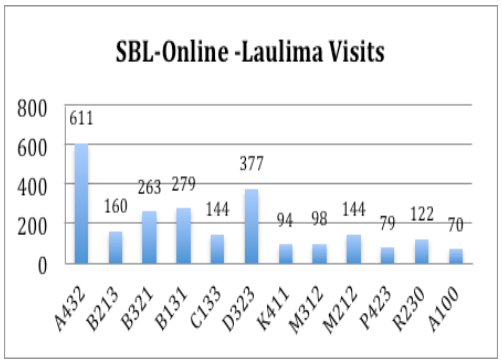
by asking each other. This conclusion was arrived to after the email count differences and after the end-of-semester questionnaire was analyzed.

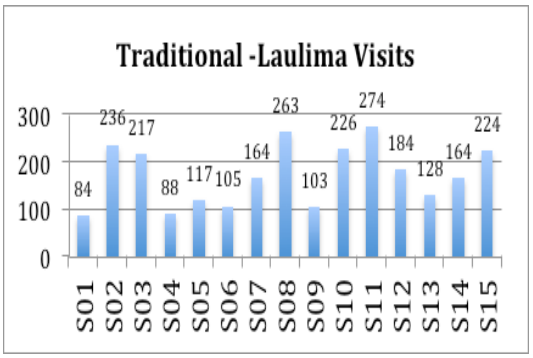
Compared to the control group there were no differences in the grades obtained during the SBL semester compared to the control group. That indicates that during the SBL semester, students decentralized the way they learn.

Laulima Thick Data Collection

Thick data collection from Laulima was used to further investigate learning differences during the two semesters encompassing this study. Laulima provides reports and statistics to address course resource utilization. Traditional students averaged 171.8 visits during the semester, while students participating in the SBL-Online had an average of 203.41. The increased number of visits by the SBL-Online participants may be attributed to several causes. In Laulima students may search for resources, send email or connect to Elluminate. Precise exact activities cannot be named.

Table 11. Laulima Visits





Common Elements Shared By This Study and the SBL Project

The SBL Project data collection requirements include the Modified MSLQ and the End-of-semester questionnaire. Both can be found in Appendix B. This section describes these elements.

Modified MSLQ

The modified MSLQ provided an inner view into student attitudes in regards to their peers as well as the course. The MSLQ sections adopted by the SBL project for this study include the following seven categories:

1. Task value
2. Self-efficacy
3. Critical thinking
4. Task goal orientation
5. Intrinsic goal
6. Sense of connectedness
7. Sense of communities of learning

The grades from the SBL semester compared to the control group are very similar and that is remarkable, given the difference in pedagogies. Student grades can be found in Appendix E. Since the grades are no different, it becomes necessary to find differences elsewhere. An important trace to follow is the great difference in the number of emails sent by students during the traditional vs. the SBL semester. If students are not emailing the instructor, it is necessary to find where were students getting the answers to their questions and how their learning process is different.

In order to find out if students were connected to each other and relied on each other instead of the instructor, it was necessary to analyze the MSLQ's sections; "sense of connectedness" and "sense of communities of learning." Students' answers are shown in Tables 12 and 13 for traditional and SBL-Online respectively. Rows highlighted in yellow represent questions worded inversely. In those cases a lower average is indicative of increased connectedness.

Table 12. MSLQ Sense of Connectedness and Community (Traditional)

Spring 2011 (Traditional)															
Sense of communities connectedness															Average
I feel that students in this course care about each other.	4	4	3	3	3	3	5	3	3	4	3	4	3	5	3.571
I feel connected to others in this course.	4	3	3	3	3	3	5	2	3	4	2	3	3	5	3.286
I do not feel a spirit of community.	1	3	4	1	2	2	1	4	2	2	4	3	4	1	2.429
I feel that this course is like a family.	3	3	3	3	3	3	5	2	2	4	2	3	4	5	3.214
I feel isolated in this course.	1	3	4	1	2	1	1	4	2	2	3	3	3	3	2.357
I trust others in this course.	4	3	4	4	4	4	5	2	2	5	2	4	4	5	3.714
I feel that I can rely others in this course.	4	4	3	4	4	3	5	2	2	3	2	4	3	5	3.429
I feel that members of this course depend on me.	4	3	2	3	3	4	1	2	2	4	2	3	3	3	2.786
I feel uncertain about others in this course.	2	4	3	2	3	3	1	4	2	2	4	3	3	1	2.643
I feel confident that others will support me.	4	2	3	4	4	4	5	2	2	5	2	3	3	5	3.429
Sense of communities Learning															
I feel that I am encouraged to ask questions.	4	5	3	5	4	5	5	4	2	4	4	4	4	5	4.143
I feel that it is hard to get help when I have a question.	1	1	2	2	3	1	1	2	2	1	4	2	3	1	1.857
I feel that I receive timely feedback.	5	4	4	5	3	5	5	4	3	5	4	4	4	5	4.286
I feel uneasy exposing gaps in my understanding.	2	3	3	1	3	1	1	2	2	1	3	3	3	1	2.071
I feel reluctant to speak openly.	2	3	3	1	3	1	1	2	2	1	3	4	3	1	2.143
I feel that this course results in only modest learning.	2	2	3	3	2	1	1	2	2	4	2	4	3	1	2.286
I feel that other students do not help me learn.	1	4	3	1	2	3	1	3	2	1	3	2	3	1	2.143
I feel that I am given ample opportunities to learn.	5	5	3	5	3	5	5	4	2	4	3	5	3	5	4.071
I feel that my educational needs are not being met.	1	3	3	5	2	1	1	2	2	1	2	2	3	1	2.071
I feel that this course does not promote a desire to learn.	1	1	3	1	2	1	1	2	2	1	1	2	3	1	1.571

Table 13. MSLQ Sense of Connectedness and Community (SBL-Online)

Fall 2010 (Studio-Based Learning Online)									
Sense of communities connectedness									Average
I feel that students in this course car	5	4	4	4	5	3	5	3	4.125
I feel connected to others in this cou	5	4	3	4	5	4	5	3	4.125
I do not feel a spirit of community.	1	2	2	2	1	3	1	3	1.875
I feel that this course is like a family.	3	3	4	3	5	3	5	1	3.375
I feel isolated in this course.	1	2	2	2	1	2	1	3	1.75
I trust others in this course.	5	4	4	4	5	4	5	2	4.125
I feel that I can rely others in this cou	5	4	4	3	5	3	5	2	3.875
I feel that members of this course de	1	3	3	2	3	3	1	3	2.375
I feel uncertain about others in this c	1	2	2	2	1	3	1	3	1.875
I feel confident that others will supp	5	4	4	3	5	3	5	3	4
Sense of communities Learning									
I feel that I am encouraged to ask qu	5	4	4	3	5	4	5	4	4.25
I feel that it is hard to get help when	1	2	2	2	1	2	1	2	1.625
I feel that I receive timely feedback.	4	4	4	5	5	4	5	4	4.375
I feel uneasy exposing gaps in my un	1	2	2	3	1	4	1	2	2
I feel reluctant to speak openly.	1	1	2	3	1	4	1	1	1.75
I feel that this course results in only	1	2	2	3	1	3	1	2	1.875
I feel that other students do not hel	1	1	2	2	1	4	1	3	1.875
I feel that I am given ample opportu	5	4	4	3	5	4	5	5	4.375
I feel that my educational needs are	1	2	2	2	1	2	1	2	1.625
I feel that this course does not prom	1	1	2	2	1	1	1	2	1.375

Based on students' responses to the sections it is highly evident that students in the SBL-Online treatment feel highly connected to their peers, they trust each other and value their peers' opinions. Students that experienced the SBL-Online have a very high sense of community where they not just feel that others can help them learn but they consider themselves pivotal to the learning of others.

Students in the control group felt that their connection to their peers was low. They exhibit some trust, and feel that they can rely in their peers, but not to the same extent of those in the SBL-Online group.

In regards to course elements delivery and instructor disposition, students feel that the instructor takes care of them, encourages questions and is available to help. In both semesters students agreed that they received timely input for their projects.

Table 14. MSLQ Attitudes in Regards to Peer Learning

Peer Learning															
Studio-Based Learning															
When studying for this course, I often try to explain the material to a classmate or a friend.	2	1	3	3	7	4	7	1	3.50						
I try to work with other students from this class to complete the course assignments.	4	2	3	5	7	5	7	2	4.38						
When studying for this course, I often set aside time to discuss the course material with a group of students from the class.	2	1	2	3	7	4	7	1	3.38						
Traditional															
When studying for this course, I often try to explain the material to a classmate or a friend.	6	4	5	5	5	5	7	3	3	5	4	4	4	4	4.57
I try to work with other students from this class to complete the course assignments.	6	2	5	5	6	4	7	3	4	6	4	4	6	5	4.79
When studying for this course, I often set aside time to discuss the course material with a group of students from the class.	5	2	5	5	6	1	7	1	3	5	3	4	5	4	4.00

These courses were delivered entirely online, so, if students meet in real life, it may be due to any number of reasons. Addressing this issue lays a question within the modified MSLQ: "When studying for this course, I often set aside time to discuss the course material with a group of students from the class," it was very surprising to find out that most of the students in the control group say that they get together with others to study whereas those in the SBL-Online group do not. This is unexpected given that according to the student's sense of community, the SBL-Online group has a much tighter community than the control group. In regards to working together to complete assignments, both groups report about the same results. Furthermore, the control group students get together to study more often than those in the SBL-Online group, but both groups spend the about the same time working on assignments together.

End of Semester Questionnaire

The end-of-semester questionnaire helped identify students' feelings in regards to the inner mechanics of the course. Just like student grades can be seen as a measure of student success, the end-of-semester questionnaire can be seen as a measure of success of the SBL-Online pedagogy. The end-of-semester questionnaire is about students' feelings and perceptions. The following will address student answers to relevant parts of this questionnaire as they relate to this study. Differences between traditional and SBL-Online are highlighted as needed. The complete end-of-semester questionnaire can be found in Appendix B.

Every student that took the end-of-semester questionnaire reported having enjoyed the course more than they initially thought. They reported that they learned more than they expected to learn. Students without programming experience affirm that this course changed the way that they approach problem solving, while students with experience reported no changes in that respect.

Figure 36 is an extract from the end-of-semester questionnaire, displaying all the answers received from both SBL-Online and traditional students to the questions:

Did you enjoy this course more or less than you expected at the beginning of the course?

Did you learn more or less than you expected at the beginning of the course?

Traditional			
1.1 Did you ENJOY this course more or less than what you expected at the beginning of the course?	1.2 Please explain why.	2.1 Did you LEARN more or less than what you expected at the beginning of the course?	2.2 Please explain why.
more	I learned quite a bit, and I was able to do much better in the course than what I thought I could do.	more	I thought I was going to learn just a very basic understanding, but we went over a lot of different methods and how they intertwine with each other.
neutral	In the beginning of the course, I really liked what I was learning because it had something to do with my major. But the last 3 assignments are too much, I don't think I'll even need it for my major.	more	I learned a lot. In the beginning I didn't know how to program. I didn't know anything about programming, but now I can honestly say that I'm pretty decent at it.
more	i learnt alot	more	i learnt moor
less	I felt that the material became harder to understand	more	The learning curve of this course was fairly large and a lot of knowledge of past assignments were useful for the next assignment that were given.
more	Because i can see how programming is very important in technology	more	The course material was a lot so i learned a lot
more	I found the java programming very interesting and challenging.	more	Java is a new language to me and I love learning.
more	learning to be meticulous in my programming.	more	If I forgot something I always looked it up or ask my classmates.
more	I didn't like programming at first. It's not so bad this time around.	neutral	I didn't really have any expectations
neutral	some of the assignment were kind of hard	more	learned how to java program
more	I relearned some material I have forgotten and got to see different views from other classmates.	more	I learned more because this class was through and went over a lot of programming methods.
more	The assignments were the most challenging and I like that it weighed more than the quizzes.	more	Approach to the material was meaningful and the assignments structured to enforce what was learned.
more	It made me think about different possibility and that I could create something out of nothing.	more	Because it takes time and a lot of thinking to complete the programs.
neutral	I learned alot more than I expected to, and while I do feel that I have learned alot, I am not completely reassured that I will keep up this knowledge.	neutral	n/a.
more	I did enjoy this course even though there were some difficulties, I liked that I could create something my friends can see online	more	I did learn more than expected because I did not know what was going to be covered
more	learned more than i thought i would	more	did not actually think I would stay in the course
Studio-Based-Learning Online			
more	Since I had done many of the programs before, it was easier to understand the second time around.	more	I could see my mistakes more than last time.
more	I really didn't know what to expect, but the class was challenging to the point of being interesting without being overly challenging.	more	I really didn't have any idea what I was getting in to, but the course covered a lot of information in a relatively short period of time.
more	I had a shoddy experience with an online "course" in the past (for a different subject) due to there being no actual instruction; we were expected to just buy a textbook, read the material and do (ungraded) assignments on our own, then be graded entirely on exams. Was a total waste of my time and money all around.	more	Same as above
neutral	I took this course as a requirement for my degree. I've done programming in the pass , so I knew what to expect.	more	It got quite involved at the end of the course
more	Being able to use illumination and communicating with others who had the same problems or questions that I might have had.	more	Extremely more. As the semester progressed the programming became intense. It was like beginning with the basics and building upon it.
more	Lot of hands-on learning/assignments.	more	Gained a lot of technical knowledge.
more	it was easier to understand that i thought it would be	more	i had a background of the subject and understand more about it now

Figure 36. Attitudes Towards the Class

The end-of-semester questionnaire also shows that roughly half of the overall students said that the video lectures made a notable difference in their learning. The ability to pause and rewind an explanation was deemed the most important feature. About 50% of the students in the SBL group, attribute their learning to the SBL-Online sessions. This can be seen in Figure 37. Reportedly, SBL-Online sessions helped students by providing different approaches to solving a problem.

In Your view, what impacted your learning more?	
Traditional	**SBL-Online**
If I was having trouble I could just watch the video again, look at the examples that were given, or consult my classmates.	It was helpful seeing how someone else did it, and having someone correct your mistakes.
Professor Blanca tried to explain everything that we needed to complete the assignments in her lectures. She also provided us with excellent examples to try and help us with the assignments.	Being able to see how other people were doing the same assignments and/or getting feedback about my code from other people really help.
i learnt how to werk wit oders.	Due to the lectures being recorded as video, it was very easy to pause/rewind and view the material at our own pace
The instructor's lecture impacted my learning, by demonstrating examples during lectures.	The only way to understand the content was through the videos
I had to pay close attention to the lectures and Ms. Polo took the time to explain the concepts and use real life examples.	If I did not understand the first time what the lectures were about. I can always return back to the ipod and re-play it again. It helped me out alot.
Because it showed step by step and examples of how to do the programs.	The actual application of programming techniques.
If I did not understand the problem I can reply back to the part where I can find my answers to my questions.	the powerpoint presentations and goole.com
i learn the most through application	
Doing the assignment made me think really hard. Some assignment I spent hours doing.	
With other students we got to see how each other did their work and learn new stlyes of coding.	
I'm assuming this means the assignments. I did meet and communicate with a few students from the class.	
Mostly the lectures because you can pause and rewind over and over again	
she further explained the material beyond what the book alone states. It really helped me to grasp each individual concept better.	
In addition to the lecture, in order to learn and make sure I understand the material, a lab would be the ieal venue to test my skills	

Figure 37. What Impacted Learning

Figure 38 displays students' opinions in regards to assignment feedback given to students by students during the SBL-online version of this course.

Traditional		Studio-Based Learning Online	
How specifically, did the feedback help you?	**Why was feedback not useful?**	**How specifically, did the feedback help you?**	**Why was feedback not useful?**
The feedback given was used to help me write the programs better, and to have a sharper eye for errors.		It showed me where my errors were.	
	I did not receive any feedback	It let me know where I was not following through with certain aspects of the code.	
to finish		When I was having a problem with an assignment, the feedback I received often clued me in immediately to what the problem was, which was usually some	
	I did not receive any feedback	I could see the errors i made.	
Positive feedback if i did really well and add some points on what i should of done		My program would be slightly different from another student and we compared	
It gave me direction when I was lost or confused.			The feedback I received was not helpful
It helped me understand what I am doing			
	I did not receive any feedback		
They help with me with my java programs when I got stuck			
It taught me how to become a better programmer and to help make the program easier for others to view.			
It is sometimes helpful to listen to how others approach problem solving. The instructor provides some hints for the assignments so that you know whether you are on the right track.			
Teacher helped me whenever I got stuck on a certain problem			
helped me to understand where I went wrong.			
I got hints and suggestions on how to make my program work			
	I did not receive any feedback		

Figure 38. The Impact of Feedback in Students

Assignment feedback given to students was useful because it provided students with hints about how to solve their problems. This is evidenced by the following messages.

> *"When I was having a problem with an assignment, the feedback I received often clued me in immediately to what the problem was, which was usually some stupid error like a missing punctuation mark. Several times in the past, I have helped others to spot a similar error in their code that they spent a great deal of time trying to locate themselves, and the same has gone for others that have*

helped me. I think it helps to develop a sense of camaraderie among programmers."

"My program would be slightly different from another student and we compared our codes and we could figure out were we went wrong."

Students concur about the fact that explaining to others helps them to better learn.

"the more I teach, the more I remember"

Providing others with feedback gives students extra understanding of the subject. It helped students grasp the different ways to code for the same outcome. It provided students with an opportunity to reflect in their own ways.

"It made me think of using the feedbacks I gave"

Students felt comfortable in regards to receiving feedback from their peers. They sensed that student's comments were well intentioned. They enjoyed seeing other student's perspectives and getting input geared towards the improvement of their assignments.

"Classmates never just made disparaging remarks for the sake of bringing me down, there was always something positive about it."

"It is okay to do something wrong and have my classmate help me out."

Students compare SBL-Online sessions to having another instructor. They enjoyed learning how others solve programming assignments. At the same time students liked the fact that they were not programming together; they were required to finish her own work before the SBL session.

"*avoided the potential problem of someone in the group dragging down group collaboration.*"

"It was like having another teacher."

"If I could understand their work, I have a better understanding of how to do my work"

Students in this group affirm that Communication is the best problem solver. It helps to pose questions because maybe other students can be wondering about the same issues. Speaking up gives everyone the opportunity to learn. Being able to contact others provides the means for learning.

"I did not know how to post my drawing on the web and I finally through via skype, email, Laulima and telephone. Communication is always there. There is no excuse of not being able to not get hold of your classmate and instructor."

Still, others don't feel quite comfortable expressing themselves. Like this student comments:

"I still feel I have to be nice"

Perspectives from the Traditional Online (Control Group)

During this traditional online implementation of ICS111, students were not required to interact with each other at all. There were no code reviews scheduled. However, 37.5% of the students reported having received help from their peers. 62.5% of the students reported having received help and feedback from the instructor, while 25% received help both from the instructor and their peers. Please refer to Figure 38.

The only evidence about student-student interaction within the control group relies in students' answers to the end-of-semester questionnaire. Here is evidence about one student meeting:

"We met for our homework Drawing.java programming. Other students scheduled a chat session where we compared our homework assignments it helped me and other students"

Not all the students were able to communicate with their peers or receive feedback from them. Students that received feedback speak highly about the positive effects that it had in their code. They acknowledge that having an extra pair of eyes looking at their code is always good. Students argue that comparing their work with that of others helps new ideas develop.

"Because the lecture the teacher provides helps solve the programs and also the teacher is there help and anytime."

"They had their own way of solving a problem and it helped me to see what were my mistakes in the way I solved mine"

Students show their kindness and willingness to help others out. Given their comments it seems that they really strive to make an impact on their peers. About half of the students that provided feedback to others found that it benefited them as well.

"One person was totally writing a program that had nothing to do with the assignment and I pointed out the goals that the program was supposed to meet. Another student was just lost and hopefully I was able to guide her in the right direction so that she could complete the assignment."

In regards to confidence, students gain confidence when they are able to understand feedback given to them, as well as when they are able to provide feedback. According to the comments in this survey, students' interactions are usually cordial.

"Most of the feedback I received from my fellow students were well met and positive, so an increase in confidence happened."

Students in the control group were asked if they wish they had worked in groups. Before addressing students' answers it is important to emphasize the distinction between studio-based learning and teamwork. Students recognize working in groups as getting together with others to build something. During an SBL session, students do not build code together, they only critique it. Because of the way that this question was worded by the SBL project, and given students' answers, I believe that students in the control group understood that working in groups meant building software together. Here is the question followed by student's responses:

Question: In your view, what impacted your learning more? [Studio/group activities (if the course had any]

"It would have been easier to complete an assignment faster"

"Would have made working on assignments easier and possibly faster."

"It would have helped with sharing opinions and ideas about projects and assignments."

"It would have been a lot helpful"

"we could solve problems faster."

"Sometimes it's very helpful to the learning process to hear how others approach the problem and how their structure their solutions."

"Would like to see the other student's point of view."

Based on the data collected, it is possible to determine that feedback from student peers is perceived as a powerful element within a programming class.

Shared Course Elements

This section compares student outcomes from the control group and the SBL-Online group. This is done first by comparing student grades, followed by a detailed analysis of student errors within quizzes and exams.

Student Performance

Analyzing student grades through the semesters in question, made evident the fact that there are not great differences between outcomes in the control group and the SBL-Online group. Table 15 shows student grades in assignment and assessments during this period.

Table 15. Comparison of SBL-Online vs. Traditional Grades

	Assignments	Quizzes	Midterm1	Midterm2	Final	Course
SBL-Online	82.01%	66.5%	77.3%	77.8%	70.7%	79.1%
Traditional	82.66%	67.3%	73.5%	70.5%	60.7%	74.0%

Table 15 shows small difference in quizzes' grades. Midterms show a moderate difference while the final exam shows clearly a better performance by students in the SBL treatment, therefore the final grade is slightly better for them as well. However, the similarity in grades obtained in assignments is remarkable given that students in the SBL-Online treatment were able to fix and resubmit their assignments.

A closer comparison was made to identify the kind of errors that students made while taking the final exam. Table 16 shows the comparison of student errors during the final exam:

Table 16. Final Exam Comparisons

SBL-Online Fall 10	shoe object	seconds conversion	price of tour		Traditional Spring11	shoe object	seconds conversion	price of tour	
Problem	F-1	F-2	F-3	total	**Problem**	F-1	F-2	F-3	total
if-else and switch. {blocks} related to selection, validation	2	0	4	6	don't understand what they have to do	2	7	1	10
don't understand what they have to do	1	3	0	4	if-else and switch. {blocks} related to selection, validation	0	1	6	7
arithmetic expressions, calculations or precedence (random)	0	3	0	3	arithmetic expressions, calculations or precedence (random)	0	3	0	3
arrays as a data structure. Subscripts, contents, declaration, initialization	0	0	2	2	related to repetition. Loop headers, bodies and blocks	0	0	3	3
braces, brackets, semi-colons, typos, spelling, naming conventions, import statements forgotten, output formatting -using magic numbers comments	0	0	1	1	basic sequential flow of control. Role of main or init methods. Call and return of methods	1	0	0	1
Exceptions, throw catch	0	0	1	1	Exceptions, throw catch	1	0	0	1
string methods	1	0	0	1	they know what to do but not how to do it	0	1		1
initialization and assignment	1	0	0	1	initialization and assignment		1	0	1
					concept and purpose	1	0	0	1

It is important to emphasize that students have been exposed the same video lectures and they have been assigned the same programming projects and assessments. Overall, the students in the SBL-Online section had a total of 19 errors while the control group had a total of 28. This is a 67.86% difference in the final exam, which is meaningful.

Reflections

Lack of statistical significance reported by the SBL Project element analysis focused the objectives of this study on what happens inside the studio. This study provided a descriptive insight on how students behave as part of a community of peer evaluators; how they feel about the process and how they feel about each other. Learning is not just about grades; it is about growing intellectually and within a community of peers.

Summary

This chapter presented the data collected during the study in such a way that it addressed the research questions. The video transcription process was revisited. The SBL-Online sessions were analyzed and commonalities and differences among the different SBL-Online rounds of sessions were identified. Students' attitudes towards the

course and each other were visualized from different perspectives allowed by the modified MSLQ, the end-of-semester-questionnaire and assignment submission messages.

Chapter 6. Conclusions and Future Work

This chapter enumerates the findings and presents the conclusions of this study. It begins by addressing the research questions that guided the development of this dissertation. It describes the unique characteristics of SBL-Online, followed by the contributions made by this study and guidelines on how to plan and deliver a successful SBL-Online implementation. Aspects of the positive outcomes and limitations of SBL-Online are discussed. Finally, this chapter concludes with reflections about this study.

As posed throughout this dissertation, the purpose of this study is to describe how SBL-Online impacts student's behavior, attitude towards the course and semester grades in an online CS1 course. As described chapter 4 and with the guidance of the data inventory in Figure 10, it was possible to address the research questions.

Addressing the Research Questions

The first research question is addressed by comparing the SBL-Online treatment with the traditional version of CS1. The remaining questions were addressed by analyzing SBL-Online video transcripts. Independently of the main source of information used to address the research questions, thick data was used to verify any conclusions drawn.

RQ1. Is there any indication how the application of SBL-Online aids a beginning programming class?

SBL-Online sessions aid a programming class by providing a forum for student interaction and exchange of ideas. The way that SBL-Online sessions are planned using previous student errors, placed within the curriculum timeline and executed online, does more than virtually placing students together at a particular time. It allows students to express themselves as well as to help each other while difficult topics are still fresh in their minds. While interacting during SBL-Online sessions, students take the opportunity

to interact with each other socially as evidenced by the great percentages of social interaction within each session.

After analyzing video transcripts it becomes evident that contributions by the instructor during SBL-Online sessions are minimal. In most of the cases instructor contributions were either a short explanation to answer questions related to the organization of work. Most of the exchanges of ideas, including explanations, helping and providing feedback as well as organizing work are student-student interaction. This kind of interaction exemplified the constructivist design model of reciprocal teaching which has been argued, benefits students because it allows the learner to interact with each other and with the environment to acquire knowledge (Jonassen et al., 1995).

As specified before, the remaining research questions are addressed by analyzing SBL-Online sessions as well as the modified MSLQ, the end-of-semester questionnaire and assignment submission messages.

RQ2. What are the interaction highlights?

Interaction highlights were addressed in detail in chapter 4. This is a summary containing the most significant interaction highlights found throughout the SBL-Online sessions:

Real-time instructor-student interaction took place within the SBL class online, during the Elluminate sessions. When the instructor entered the Elluminate virtual room students usually asked questions about the process and deliverables. Questions related to course topics were few. However, a great number of times the instructor was ignored or quickly dismissed when entering the Elluminate environment. Students were deeply engaged in the review process, as it can be seen in most session transcripts available in Appendix E.

When facing technical difficulties, students demonstrated cohesiveness and determination. They were resourceful and found ways to get the work done.

During the sessions, students were verbal with each other about their thoughts and knowledge, or lack of it. Students carefully listened to each other's feedback acknowledging or asking questions to verify that they understood it.

In regards to teaching/learning interaction, students did not usually ask for help but instead they asked for feedback. Students showed genuine care for each other. As it becomes evident in the end-of-semester questionnaire students felt that they were pivotal in the learning of their peers. Some students enjoyed mentoring their peers, while the mentees were appreciative of the help they received from their peers.

Students got along well; they showed respect for each other. Overall students in the SBL treatment expressed having enjoyed the class and getting to know their peers. Peer mentoring is a valuable resource, seldom utilized.

RQ3. Is there any indication that learning is taking place?

The purpose of this study is to describe how SBL-Online impacts student's behavior, attitude towards the course and semester grades in an online CS1 course.

In regards to student learning in an online environment, independently of pedagogy, it has been said that:

> While conferencing, the learner is electronically engaged in discussion and interaction with peers and experts in a process of social negotiation. Knowledge construction occurs when students explore issues, take positions, discuss those positions in an argumentative format, and reflect on and re-evaluate their positions. (Jonassen et al., 1995, p. 16)

This study has shown high levels of engagement, from part of the students as they discuss course topics and evaluate peer assignments during SBL-Online sessions. Therefore it can be implied that knowledge construction has taken place. However, this may not be enough evidence to claim that learning has taken place.

Following the previous premise, learning should be identifiable within student-to-student assistance, or peer teaching. The search for learning initiated with all the codes grouped under "contributions. These are contributions to the learning of others, which may or not be prompted. Learning moments were identified by means of the following equation:

Learning Moment =

Any one of the following codes, FBS|HES|ADE|EKE|COM|SDR|TLR]

Followed by any one of these codes [HEG|FBG|SKN]

Any of those interaction pairs may denote a learning moment. Note that this does not imply a large body of knowledge being assimilated by the students at once, it signifies a little bit of knowledge has been transferred. It may also be the validation of knowledge. Pattern identification and counts were done with the use of a programming script. The script can be found in Appendix C. Outcomes revealed that other learning patters are amiss because in some sessions, the learning moments do not reflect back on the code counts. For example, in session2-team3, the identifiable learning moments were only five while feedback seeking (FBS) and feedback giving (FBG) top the code counts.

As evidenced in Table 7, Help Seeking (HES) appears in 80% of the sessions in the lower half of the code count, while feedback seeking (FBS) is 90% of the time in the top-three code count. This evidence showed that students are not likely to ask for help, while they are very likely to request feedback. This fact indicates that students have acquired the basis to program the required assignments and request feedback to finish or correct them.

RQ4. Are students' misconceptions being identified and addressed?

As described in chapter five, misconceptions can arise in various forms.

- Hypothesis (HYP), when a student wrongly expresses her beliefs about a situation.
- Explaining known errors (EKE), which arise when students explain what is wrong with the way their code behaves, and they do not know the cause.
- Comparisons (COM), when a student compares coding structures, behaviors and outcomes of two programs and the comparison is flawed.
- Sudden realization (SDR), when a student suddenly realizes something that was creating a hole in his body of knowledge.
- Thinking out loud and reflecting (TLR), when students candidly express an inaccurate thought in regards to the code process.

In most of these cases, misconceptions are addressed either by students giving feedback (FBG), sharing knowledge (SKN) or helping each other (HEG). Misconceptions were shared more than identified. Students addressed misconceptions in nearly every occasion. The instructor clarified only a handful of them.

The best evidence of misconceptions being addressed lays on assignment resubmission as well as submission messages. As explained in chapter five, each one of the assignments resubmitted were improved versions of students first submissions. Occasionally students did not resubmit their assignments. In those instances it was not possible to determine if misconceptions were addressed or impasses were conquered. Overall, it can be said that during and after every SBL-Online session, misconceptions were addressed and errors eliminated.

RQ5. Are students acquiring knowledge that will transfer into other course activities?

Transfer of knowledge was addressed by looking into the SBL-Online sessions transcripts to find if there was any evidence of the knowledge acquired in this class being transferred into other areas of the curriculum. The initial process was to find it within peer-to-peer social interaction (SI). It was expected that students would talk about transfer of knowledge casually. However, what transpired during social interaction were jokes and funny comments that spiced up the sessions, nothing regarding course skills.

While other studies have compared the transfer of knowledge as the knowledge taken from one class to the next (Hundhausen et al., 2008), this study takes a different approach, an approach that begins and ends within this study. When looking into reflections about course work (RCW), it was clear that RCW is advice given from peer to peer. It transfers course-related knowledge and it is a way in which students reflect out loud about the course at the same time as they help each other. Transferrable knowledge exchanged is not usually about programming or course topics but is instead about survival and tips to get a better grade in the class.

Unique Characteristics of SBL-Online

The main characteristic of SBL-Online is that it implements the studio-based learning pedagogy. However, what makes SBL unique is the fact that it was built based on errors that students made in previous semesters, and that it uses these errors to focus SBL

sessions into addressing those errors. Lastly, SBL-Online implementations are done fully online.

In respect to online education, it has been argued by Moore (1989) that student-student interaction greatly benefits student learning. SBL-Online promotes this kind of interaction with the added benefit of focusing students' attention in recurring errors with the goal to break the recurrent error cycle.

Student-Student Interaction

According to Barker (2009), student-student interaction is a powerful predictor of students' intentions to persist in the major. Promoting interaction during the course of the semester should be expected to increase retention. This has been shown in SBL-Project longitudinal studies but cannot yet be shown in this study.

It was mentioned several times during the course of this dissertation that students may meet outside the online classroom. At the beginning of the semester, students were informed that they could email each other using Laulima. Laulima has an email option where students can click in the role of "student" and they can see all the students' names and their email addresses. Students were also informed that they could use Elluminate to virtually meet with each other without the need for the instructor to be there. Students in both iterations of the course were introduced to Elluminate in an "ice-breaker" session.

During the SBL-Online semester student-student interaction is part of the requirements to pass the course. Students must interact in teams at least three times during the semester when attending SBL-Online sessions. Student-student interaction was thoroughly documented in the SBL-Online session transcripts.

Social Interaction During SBL-Online

B321: sorry, road runner owned me
D323: I think so
D323: thanks you guys and lady of the house
IM: I hope you had fun
M312: it was nice
B321: D I need advice!
IM: I hope it was useful too
D323: sure what on?
B321: civil
M312: engineering
D323: email, Skype me
[exchanging Skype details]
B321: Who is XX?
D323: X?
B321: X.X
B. Added me
B saying
B321: in my ics class
D323: a guy in our class
B321: Xxxx Xxxx.
B321: student ICS111
B321: oh I see
D323: ok thanks for the session
B321: thanks everyone
B321: see you next time
B321: best hour of my day
D323: technology is getting quite fancy
IM: thanks
D323: bye!

Figure 39. Social Interaction

Students seemed to enjoy the time that they spend "together" during SBL sessions. It is evidenced by the amount of social interaction that takes place. In fifty percent of the sessions, social interaction was part of the top three code counts (please refer to Table 10). Students take time to joke as they learn.

Social interaction is not just about jokes and small talk, it is about building relationships and enhancing camaraderie. This can be seen at the end of Session1-Team 3, where students spent the last five minutes of the session talking about topics external to the class, and asking each other for advice.

Peer Reviews

According to Hundhausen et al. (2009) students prefer reviews from other students with less experience than themselves. Even in such a case, peer reviews are accurate compared to an accepted evaluation standard. They also argue that participating in peer reviews results in better learning outcomes. To exemplify that, this is a face-to-face comment given by a student that experienced the SBL-Online: "You need to really understand what is going on in order to provide your classmates with adequate feedback."

Peer reviews are like real-world industry code reviews (Hundhausen et al., 2009). Students therefore develop a collegial relationship with one another. Participating in peer reviews tacitly assumes that students are prepared. As mentioned before, explaining and learning the thought process invested into product creation helps students learn.

Students prefer to evaluate each other's work instead of building something together. This way they do not depend on others for their grade. This is a direct quote by an SBL participant, taken from the end-of-semester questionnaire:

> The group work consisted merely of review/feedback sessions, which I liked, because everyone still had to complete their own work and it avoided the potential problem of someone in the group dragging down a group collaboration (hate to say it, but it often does happen).

Data from this study indicates that as students become more comfortable with SBL they become more efficient as peer evaluators. Students are able to critique and to provide feedback in a shorter amount of time. However, when they become too efficient and dedicate more time to the mechanics of producing deliverables, they learn less and produce more.

To measure the effectiveness of pedagogical code reviews (PCR), Hundhausen, Agarwal, and Trevisan, 2011 investigated the effect that PCRs would have on a student's attitudes through a modified version of the Motivated Strategies for Learning Questionnaire (MSLQ) and they noted that PCRs promoted an increase in attitudes toward peer learning. These results match those found during this study where students feel as they are part of a community where everyone plays a role in everyone else's learning. In this study, results are similar to those found by Estey et al. (2010).

Errors

Student errors played a large role in the setting of this study. Errors were collected, quantified and localized within the curriculum timeline in order to properly set this study. However, the implementation of SBL-Online did not reduce the initial amount of errors students made. Students tend to make the same mistakes as their predecessors. However, if error counts are performed within assignment resubmissions, error counts are negligible as evidenced on chapter five.

In most cases, turning in an assignment usually means moving on to the next topic. This is especially detrimental when a student turns an incomplete or faulty assignment. This kind of situation does not provide a sense of accomplishment in the students' leaning process, and it reflects in students' assignment submission messages labeled as "terminal." A full list of terminal messages can be found in Appendix F.

Assignment Submission Messages

Assignment-submission messages from students are not required. Given the kind of submission messages that were collected, it has been noted that the ability to comment within assignment submission gives student closure within each assignment. Overall, there are very few students that do not comment when submitting an assignment. This fact was surprising as well as insightful and useful to this study. Student's state of mind after completing an assignment could be directly assessed based on submission messages.

SBL-Online to Help a Beginning Programming Class

When comparing the effectiveness between SBL and traditional instruction, Hendrix et al. (2010) found out that both the traditional and the SBL implementations were both sufficient at delivering the basic course content. It is arguable that it is not the destination that matters but the journey. Given the thick data collection performed in this study, it is evident that students in the SBL-Online treatment found varied ways of obtaining the same knowledge and perhaps in more depth that those in the traditional settings; even when there are no considerable differences in student grades. Yet, there are noteworthy differences in the learning path that students navigated through the semester.

Students in the control group gravitated towards the instructor for help. During the traditional semester, several students regularly attended online as well as real office hours. That was not the case during the semester where the SBL-Online was implemented. Definitely, students in the SBL-Online iteration did not contact the instructor as much as those in the traditional group. The difference in email counts is almost three-fold as seen in Table 10. Students in the SBL-Online iteration visited the Laulima site much more than the control group. There is a limited kind of information in regards to what students do while they are in Laulima and it is not precise enough to draw any valid conclusions. However, evidence suggests that they learned to make better use of the available resources, and those resources did not include the instructor's time. This may be an indication that SBL promotes student resourcefulness.

Contributions Made by This Study

This study made several noteworthy contributions to the field of computer science education. These contributions include the concept of proactive teaching, the implementation of SBL pedagogies in a fully online course, a code for video transcription to better reflect activities related with SBL and peer-to-peer teaching and learning and lastly it provided an inner view to the inside mechanics of an SBL-Online session.

The Formal Conceptualization of Proactive Teaching

Proactive teaching as presented and implemented within this study refers to an educational approach geared towards applying solutions where common problems are known to arise. Proactive teaching can be generalized and used in any discipline. Proactive teaching is based on student errors that are known to occur. These errors should be identified, counted and categorized. With this information on hand assignments and extra course activities should be given to students so that they can overcome these errors and the limitations that they impose in their learning.

Proactive teaching can be seen as a great tool for online course delivery, where the instructor has little or no opportunities to patch student knowledge after a failed assignment or assessment.

The Implementation of SBL Pedagogies in a Fully Online Course

CS1, which is the course object of this study has been both taught on line and has been taught implementing SBL pedagogies. However, both approaches had not been done simultaneously within one course. SBL-Online represents the united application of these two approaches into a CS1 course. Aside from that, SBL-Online pursues the goal of helping online students overcome impasses product of recurring errors within the curriculum.

Being that the course is online and with the use of multimedia technologies it was possible not just to implement, but to also record student-student interaction taking place during SBL sessions. Being student-student interaction one of the major predictors of student retention (Moore 1999), the application of SBL in a CS1 increasing student-

student interaction may result in student retention. Retention information from this courses, however, is not available yet due to the fact that this study is still very recent.

The Adaptation of a Transcript Code to Address SBL-Online Interaction

The code produced by Curtis and Lawson (2001) to transcribe online collaboration was modified and adapted to address the particulars of SBL-Online interaction. Code modifications were made with the purpose of addressing the research questions posed at the beginning of this study. Codes added mainly address the different kinds of utterances that students use when seeking of giving help or feedback.

The modified code for SBL-Online interaction is to be used in context and a full explanation of how this is done can be found on chapter five. This code is what makes the identification of "Learning moments" possible.

The Identification of Patterns That May Indicate a "Learning Moment"

As explained at the beginning of this chapter, a learning moment can be defined a predefined sequence of codes within an SBL-Online session transcript. This work has identified patterns believed to signal learning moments. Searching for these patterns, is expected to determine the amount of learning that took place during a given SBL-Online session. However, the researcher believes that work in this area needs to broaden in order to identify additional patterns that may also give way to learning moments that may have been missed by this study.

An Inner View into the Mechanics of SBL Interaction

Given that the class in which the SBL pedagogy was implemented was online and with the help of the Elluminate software it was possible to record the process that students underwent during each one of the SBL sessions. Because Elluminate's cameras are not visible, since they are internal to the process and software-based, students did not physically see them, consequently, students relaxed. This provided a more natural environment for students during SBL-Online sessions. This is evident in the session transcripts, which can be found on Appendix B.

Guidelines to a Successful SBL-Online Implementation

The contributions listed above can be greatly useful for anyone wishing to implement SBL techniques in an online environment. For this, it will be necessary to begin by becoming officially aware of recurring errors within the target course by collecting recurring errors made by students taking the same course where SBL-Online is going to be implemented.

Once recurring errors are identified it is necessary to locate the points in the curriculum where topics related to those errors are introduced. The expectation is that errors will form visible clusters. Those clusters will mark the possible times within the semester where SBL-Online sessions should be scheduled.

After identifying times in the curriculum for SBL sessions to take place it is necessary to create assignments that will force students to implement elements that are known to cause the target errors.

When no official clusters of errors have been collected or identified, SBL-Online can still be implemented. The way to do it will be based on instructor's observations. Based on experience, the instructor can make an educated guess in regards to topics that she perceives are hard for students to learn. Preemptive teaching can then be applied centered around those topics. Instructors can refine error clusters semester after semester until clusters of errors are officially identified and SBL-Online is fully implemented.

There is a downside to SBL-online implementations. If a course has no clusters of errors, that is, many different errors are found but no clusters of them, perhaps that course is not one in which SBL-Online will make a significant difference in student learning. SBL-Online requires of clusters of errors to guide the content and timing of the sessions.

Reflections About This Study

There is a discrepancy between the SBL project data collection and the data collection made for this study. The SBL project only reports results of students that answered all the SBL-Project required elements. For example if a student partially participated, that student's data would have been omitted from the project. For this study, a thick data collection took place. It included data of all participating students including those that dropped out. If they participated in one SBL-Online session their data became part of this study.

From the planning of this study and as described in chapter three, the data inventory was divided into three parts to address the two sides of this study. Figure 9 explains how the data inventory addressed this dichotomy.

The data collection required for the SBL project allows the comparison of the traditional vs. the SBL-Online iterations of this course. The thick data collected helped measure the outcomes from a different perspective. SBL-Online sessions were recorded, transcribed and codified. This allowed an insight into the inner workings of a SBL-Online session. The thick data collection provided a deeper insight into students learning experiences as they reflect within their assignment submission messages. It is very meaningful to see that the data elements collected are in harmony with each other.

In regards to the session recordings; even when students are being recorded and they know it, they act as if they were not being observed. It may be because they do not see a camera. This is an advantage of the medium. Elluminate records the session without invading student's personal space.

In regards to the generalization of data, since most of the traditional vs SBL comparisons matched the data found by this study, it can be said that the data yielded by this study in regards to the inner workings of the SBL-Online can be applicable to other SBL implementations.

Conclusions

In Chapter 1, it was stated that this study was expected to create new knowledge in regards to applying SBL techniques in the online environment, as well as getting an inside view on how students interact as they showcase their work and attempt to understand other students' thought processes.

Moreover studio-based learning pedagogies in the form of SBL-Online can be successfully applied to online courses, taking advantage of the medium. That gives students the freedom of showing weakness in front of their peers, as well as the ability to give and receive feedback and constructive criticism.

Feedback is only good if it is applied. Outcomes of this study show that students get closure when they attend SBL online sessions. They are able to conquer required knowledge before moving on to the next assignment. Since SBL was first applied into online ICS111 there is a feeling that when SBL is not applied, students are not maximizing their learning potential. The lack of student-student interaction robs students from a great learning opportunity.

SBL makes students more resourceful. In order to maximize the SBL experience, it is important to acknowledge student errors in order to proactively teach online.

A successful student is not determined by the grade she gets at the end of the semester but about learning to learn about a particular subject. Students do not need to know everything but they need to count with the resources to find anything they need. An effort is made to provide students with the basic knowledge to filter good information from noise, in the hopes that students will become resourceful critical thinkers. This research shows that SBL students acquire skills, which they will be able to practice for life.

Appendix A. Assessments

Midterm Exam 1

Worth 10% of the total course grade

(15 points)

A good employer aware of the country's bad economic situation decides to give their employees a 5% raise. Write a program that will take the employer's yearly pay as input, then it will add 5% to it. After this, it will divide it by 12 to get the monthly payments. Your program should print the following information:

- original yearly salary
- the 5% raise
- new yearly salary
- new montly payment

(15 points)

Write a program that will generate random numbers as follows:

- A number between 107 and 200
- a number between -20 and 0
- a number between -15 and 25

(5 points)

I have the following snippet of code:

```
int i1 = 25;
int i2 = 26;
double d = 25.99;

i2 = (int)d;
```

The question is: Would i1 == i2 be true or false?

A. **true** (*)

B. false

(5 points)

If you didn't have the **System.out.println( )** instruction and you only have **System.out.print( )** How would you perform the same functionality of System.out.println( ) with System.out.print( )?

(5 points)

You have the following declarations:

```
int x = 29;
int y = 32;
int z = 4;
int res = 0;

res = x/z;
System.out.println(res);
```

What will be printed?

A. 7.25

B. **7** (*)

C. 8

(5 points)

You have the following declarations:

```
int x = 29;
int y = 32;
int z = 5;
int res = 0;

res = x%4;
System.out.println(res);
```

What will be printed?

A. **1** (*)

B. 7.25

C. 7

D. 8

(5 points)

You have the following declarations:

```
int x = 29;
int y = 32;
int z = 5;
int res = 0;

res = x/5;
System.out.println(res);
```

What will be printed?

A. 4

B. **5** (*)

C. 3

D. 5.8

(5 points)

In order to create a constant, you would use which of the following Java reserved words?

A. private

B. static

C. int

D. final (*)

E. class

(5 points)

Consider the double value likelihood = 0.013885. What would be output if DecimalFormat dformatter = DecimalFormat("0.00##"); and you executed System.out.println(df.format(likelihood)); ?

A. 0.013885

B. 0.0139 (*)

C. 0.014

D. .0139

E. .014

(5 points)

If x is the String "Hi There", then x.toUpperCase().toLowerCase(); will return the original version of x.

A. true

B. false (*)

(5 points)

If you want to output a double so that at least 1 digit appears to the left side of the decimal point and exactly 1 digit appears to the right side of the decimal point, which pattern would you give a DecimalFormat variable when you instantiate it?

A. **"0.0"** (*)

B. "0.#"

C. "0.##"

D. "#.#"

Midterm Exam 2

Worth 10% of the total course grade

(2 points)

Assume that you are defining a class and you want to implement an ActionListener. You state addActionListener(this); in your class constructor. What does this mean?

A. The class must import another class which implements ActionListener

B. The class must define the method actionPerformed (*)

C. The class must define the method ActionListener

D. The class must define the method actionPerformed in an inner class named ActionListener

(2 points)

In order to allow the user to perform an action when clicking a Button, you would add a(n) _____ to the Button.

A. shortcut

B. KeyListener

C. mnemonic

D. ActionListener (*)

E. Choice

(4 points)

Why is that in applets we do not need loops to have our program run more than once?

Hint: *it has to do with the actionPerformed method function*

(2 points)

Write a for loop that will never execute. You do not need to write the inside of the loop since it will never actually get there, you just need to write the header.

(2 points)

Which Layout Manager is used by default if you do not specify a Layout Manager for your GUI container?

A. FlowLayout (*)

B. BorderLayout

C. GridLayout

D. BoxLayout

E. TabbedLayout

(2 points)

Which Layout Manager type would you use if you want GUI components to be placed at the North, South, East, West and Center of the container?

A. BorderLayout (*)

B. FlowLayout

C. GridLayout

D. BoxLayout

E. GeographyLayout

(20 points)

Write a piece of a program that will read the day and month of somebody's birthday and will determine their zodiac sign. Both the month and the day should be read as integer numbers. Verify input by using try/catch. Verify number ranges using if statements. When the time comes to do the comparisons, you do not need to do them all, just for a couple of zodiac signs. If you have the right idea, I will know. Hint: Use both switch and if statements

- Aries - March 21 - April 20 (3/21 to 4/20)
- Taurus - April 21 - May 21 (4/21 to 5/21)
- Gemini - May 22 - June 21 (5/22 to 6/21)
- Cancer - June 22 - July 22 (6/22 to 7/22)
- Leo - July 23 -August 21 (7/23 to 8/21)
- Virgo - August 22 - September 23 (8/22 to 9/23)
- Libra - September 24 - October 23 (9/24 to 10/23)
- Scorpio - October 24 - November 22 (10/24 to 11/22)
- Sagittarius - November 23 - December 22 (11/23 to 12/22)
- Capricorn - December 23 - January 20 (12/23 to 1/20)
- Aquarius - January 21 - February 19 (1/21 to 2/19)
- Pisces - February 20- March 20 (2/20 to 3/20)

(20 points)

You will write a program that will read a number from the user. You may use either the Scanner of the BufferedReader. You will verify that the number is indeed a number (use try/catch), if it is not a number you will end the program. If it is a number then you will verify that is a positive integer in the range from 1 to 9 (inclusive). If the number is not in range you will end the program. If the number is in range you will print the number itself as many times as the number. Example: If the user enters the number 6, the output of your program should be:

666666

Final Exam

Worth 15 % of the total course grade

(3 points)

If s is a String, and s = "no" ; is performed, then s

A. stores the String "no"

B. references the memory location where "no" is stored (*)

C. stores the character 'n' and 'o'

D. stores an int value that represents the two characters

E. stores the character n and a reference to the memory location where the next character, o is stored

(3 points)

If an array is of size N, then it has positions that range....

A. from 0 to (N-1) (*)

B. from 1 to N

C. from 0 to N

D. from 1 to (N-1)

(3 points)

To find out the size of an array iArray we use the following instruction:

A. int iSize = iArray.length; (*)

B. int iSize = iArray.length( );

C. int iSize = iArray.size( );

D. int iSize = iArray.getLength( );

(2 points)

If a method does not have a return statement, then

A. It will produce a syntax error when compiled

B. It must be a void method (*)

C. It can not be called from outside the class that defined the method

D. It must be defined to be a public method

(2 points)

The behavior of an object is defined by the object s

A. instance data

B. constructor

C. visibility modifiers

D. methods (*)

E. instance data, constructor, visibility modifiers and methods

(2 points)

The relationship between a class and an object is best described as

A. classes are instances of objects (*)

B. objects are instances ofc) objects and classes are the same thing classes

C. objects and classes are the same thing

D. classes are programs while objects are variables

E. objects are the instance data of classes

(2 points)

All primitive data types have a wrapper class except for

A. char

B. boolean

C. int, float and double

D. all primitive data types have wrapper classes

(2 points)

Wrapper classes are used so that a primitive data type like an int or double can be treated as

A. a constant

B. a global variable

C. an object (*)

D. an alias

E. a static instance data

(12 points)

Write a class called Shoe.java. This class will have the following instance variables:

Color -String that must be at least 3 non-blank characters long

Size - integer must be between 5 and 10 inclusive. (to make it easier)

Price - double must be non-negative.

You will create a class Shoe.java with all its get and set methods along with a constructor and a toString Method. Be sure to use the ShoeException class to prevent invalid objects from being created.

(12 points)

Assume a total number of seconds is stored in a variable named seconds. Write a code segment to display the number of complete hours, complete minutes in the remaining portion of an hour, and the remaining seconds in the remaining portion of a minute which is equivalent to the original value stored in seconds.

(12 points)

Write a code segment, which accepts the price of a tour. A tour is priced between 29.95 and 249.99. Your code should handle any user input nicely and should continue to repeat until a valid price is entered. You should be able to handle any input gracefully.

Appendix B. Instruments for Gathering Data

Consent Form

INFORMED CONSENT for a research study entitled
"SBL-Online, Broadening Studio-Based Learning in Computing Education"

You are invited to participate in a research study to evaluate a new way of teaching computing, called studio-based instruction, and to compare it with traditional ways of teaching. You were selected as a participant because you are enrolled in ICS 111 during the Fall 2010 semester.

What is involved?

You will be asked to complete an online survey (approx. 15 minutes duration) at the beginning and again at the end of the course using an ID number given to you. In addition, you will be asked to complete a short web-based questionnaire at the end of the course. The data collected from both the surveys and the questionnaire will be anonymous and saved at a server at Auburn University. Your total time commitment will be approximately 1 hour during the term. The instructor of the course may see data collected from you, but only after your name or any other identifying information has been removed, and only after grading of this course is completed. So your answers to the survey and interview questions WILL NOT influence your grade in this course. Your course grade will be decided based only on graded items as described in the course syllabus. By agreeing to participate in this research study you are consenting to allow us to use your responses to the surveys as well as your grades and the revision of the Elluminate recordings for research purposes. This is the list of the data that will be collected.

1. Informed Consent
2. Student Pre-Survey
3. Student Post-Survey (identical to the pre-survey)
4. Student End-of-term Questionnaire
5. Student Course Grades
6. Instructor Experience Report
7. Elluminate Recordings (voice and computer screen only).

Are there any risks or discomforts?

The risks associated with participating in this study are minimal or non-existent.

Are there any benefits to yourself or others?

By participating in the study, you can expect to gain a better understanding of the course materials, and about your own attitudes and motivation toward learning. But we cannot promise you that you will receive any or all of these benefits.

If you change your mind about participating, you can withdraw permission to use your data at any time during the study. Your participation is completely voluntary. If you choose to withdraw, your data will be withdrawn as long as it is identifiable. Your decision about whether or not to give permission to use the data will not jeopardize your future relations with those who are conducting this study and their academic departments, or with the instructor of this course.

Your privacy will be protected. Any information you submit online in connection with this study will remain anonymous. Any means of connecting you to data (i.e. the list of names and ID numbers) will be destroyed. Anonymous data may be kept indefinitely, shared with other researchers at participating institutions and the National Science Foundation, used to fulfill an educational requirement, published in journals, or presented at meetings.

Recordings of your teamwork

Teamwork done using the Elluminate Live! is kept in the Elluminate server. You will log into Elluminate under a different name to protect your privacy. Session tapes will be reviewed as part of the research. These recordings will not include your face, only your voice along with screen shots. Tapes are kept in the Elluminate server for a year and then automatically destroyed. The use of Elluminate is common practice in higher-education institutions. The recordings will not be shared with anyone outside this project.

If you have questions about this study, please ask them now or contact Dr. M. B. Ogawa at ogawam@hawaii.edu. Please print a copy of this form for your records.

If you have any questions or concerns about your rights as a research participant please contact the UH Committee on Human Studies (CHS) at (808) 956-5007 or uhirb@hawaii.edu.

HAVING READ THE INFORMATION PROVIDED, Clicking the "YES, I ACCEPT" button you indicates your willingness to allow us to use your data for research purposes, and you will be redirected to the research site, otherwise you will be redirected to the course's website.

Being this, an Online course, the Informed Consent form was given via a website.

- Clicking on the "YES, I ACCEPT" button, will take participants directly to the survey that they will be completing.
- Clicking on the "No, Thank you" will take them to the course website.

Modified MSLQ: Motivated Strategies for Learning Questionnaire

Items followed by (R) are reverse-coded.

Motivation Set

Task Value

1. I think I will be able to use what I learn in this course in other courses.
2. It is important for me to learn the course material in this class.
3. I am very interested in the content area of this course.
4. I think the course material in this class is useful for me to learn.
5. I like the subject matter of this course.
6. Understanding the subject matter of this course is very important to me.

Self Efficacy

1. I believe I will receive an excellent grade in this class.
2. I'm certain I can understand the most difficult material presented in the readings for this course.
3. I'm confident I can understand the basic concepts taught in this course.
4. I'm confident I can understand the most complex material presented by the instructor in this course.
5. I'm confident I can do an excellent job on the assignments and tests in this course.
6. I expect to do well in this class.
7. I'm certain I can master the skills being taught in this class.
8. Considering the difficulty of this course, the teacher, and my skills, I think I will do well in this class.

Learning Strategies Set

Critical Thinking

1. I often find myself questioning things I hear or read in this course to decide if I find them convincing.

2. When a theory, interpretation, or conclusion is presented in class or in the readings, I try to decide if there is good supporting evidence.
3. I treat the course material as a starting point and try to develop my own ideas about it.
4. I try to play around with ideas of my own related to what I am learning in this course.
5. Whenever I read or hear an assertion or conclusion in this class, I think about possible alternatives.
6. When studying for this course, I often try to explain the material to a classmate or a friend.
7. I try to work with other students from this class to complete the course assignments.

Peer Learning

1. When studying for this course, I often set aside time to discuss the course material with a group of students from the class.

Performance Set

Task Goal Orientation

1. I like school work that I'll learn from even if I make a lot of mistakes.
2. An important reason why I do my school work is because I like to learn new things.
3. I like school work best when it really makes me think.
4. An important reason why I do my work in school is because I want to get better at it.
5. I do my school work because I'm interested in it.

Intrinsic Goal

1. In a class like this, I prefer course material that really challenges me so I can learn new things.
2. In a class like this, I prefer course material that arouses my curiosity, even if it is difficult to learn.
3. The most satisfying thing for me in this course is trying to understand the content as thoroughly as possible.
4. When I have the opportunity in this class, I choose course assignments that I can learn from even if they don't guarantee a good grade.

Classroom Community Set

Sense of Communities Connectedness

1. I feel that students in this course care about each other.
2. I feel connected to others in this course.
3. (R) I do not feel a spirit of community.
4. I feel that this course is like a family.
5. (R) I feel isolated in this course.
6. I trust others in this course.
7. I feel that I can rely others in this course.
8. I feel that members of this course depend on me.
9. (R) I feel uncertain about others in this course.
10. I feel confident that others will support me.

Sense of Communities Learning

1. I feel that I am encouraged to ask questions.
2. (R) I feel that it is hard to get help when I have a question.
3. I feel that I receive timely feedback.
4. (R) I feel uneasy exposing gaps in my understanding.
5. (R) I feel reluctant to speak openly.
6. (R) I feel that this course results in only modest learning.
7. (R) I feel that other students do not help me learn.
8. I feel that I am given ample opportunities to learn.
9. (R) I feel that my educational needs are not being met.
10. (R) I feel that this course does not promote a desire to learn.

End-of-Semester Questionnaire

Items followed by (SBL-O) were given only to students in the SBL-Online treatment group.

1.1 Did you ENJOY this course more or less than what you expected at the beginning of the course?
1.2 Please explain why.
2.1 Did you LEARN more or less than what you expected at the beginning of the course?
2.2 Please explain why.
3.1 In your view, what impacted your learning more? [the lectures]
3.1 In your view, what impacted your learning more? [studio/group activities (if the course had any)]
3.1 In your view, what impacted your learning more? [the lab (if the course had a lab)]
3.2 Please explain how.
4.1 Overall, did you enjoy your experiences in this course?
4.2 Which experiences?
4.2 Why not?
5.1 Having taken this course, are you looking forward to taking more courses in computing?
5.2 Why?
5.2 Why not?
6.1 Has this course increased or decreased your interest in the topic of computing?
6.2 Specifically, what parts of the course increased your interest?
6.2 Specifically, what parts of the course decreased your interest?
6.2 Please explain why.
7.1 Has taking this course changed the way you approach solving computing problems?
7.2 Why?
7.2 Why not?
8.1 Has taking this course changed your confidence level in solving computing problems?

8.2 How?

8.2 Why not?

9.1 Did you receive helpful feedback on your projects/assignments/labs from others?

9.2 Was it from your instructor, the TA, or other students? [Instructor]

9.2 Was it from your instructor, the TA, or other students? [Teaching Assistant]

9.2 Was it from your instructor, the TA, or other students? [Other students]

9.3 How specifically, did the feedback help you?

9.2 No, because

9.3 What kind of feedback WOULD have been helpful?

10.1 Has it been easy in this course to learn about how other students solved programming projects/assignments/labs?

10.2 Have you found it helpful?

10.2 Why not?

10.3 How did you find it helpful?

10.3 Why not?

11.1 Did you review and provide written feedback on other students' projects/assignments/labs?

11.2 What kind of feedback did you provide? or Give an example.

11.2 Why not?

11.3 Did you find that the process of reviewing & providing written feedback on others' work helped your learning in this course?

11.4 How?

11.4 Why not?

12.1 Has taking this course changed your COMFORT level in receiving feedback from other students about your work?

12.2 Please explain why.

12.3 Has taking this course changed your CONFIDENCE about the quality of the feedback you receive from other students?

12.4 Please explain why.

12.5 Has taking this course changed your COMFORT level in giving feedback to other students about their work?

12.6 Please explain why.

12.7 Has taking this course changed your CONFIDENCE about the quality of the feedback you give to other students?

12.8 Please explain why.

13.1 Has this course changed your comfort level in working with your classmates in solving problems?

13.2 Please explain why.

14.1 Did you receive helpful feedback during your group activity?

14.2 Was the feedback from your instructor, the TA, or other students? [Instructor]

14.2 Was the feedback from your instructor, the TA, or other students? [Teaching Assistant]

14.2 Was the feedback from your instructor, the TA, or other students? [Other students]

14.3 How, specifically, did the feedback help you?

14.2 No, because

14.3 What kind of feedback WOULD have been helpful?

15.1 Did you respond to feedback during your group activity? (SBL-O)

15.2 How? (SBL-O)

15.2 Why not? (SBL-O)

16.1 Did you find being an audience member when other students presented their work helpful to your learning in this course? (SBL-O)

16.2 How was it helpful? (SBL-O)

16.2 Why was it not helpful? (SBL-O)

17.1 Did you ask questions and/or provide feedback during other students' presentations? (SBL-O)

17.2 When/what part of student presentations prompted you to provide feedback? or Please give an example. (SBL-O)

17.2 Why not? (SBL-O)

18.1 Did you find asking questions and/or providing feedback helped your learning in this course? (SBL-O)

18.2 How? (SBL-O)

18.2 Why not? (SBL-O)

19.1 Did you work in groups? (SBL-O)

19.2 Did you like working in groups? (SBL-O)

19.2 Do you wish you had worked in groups? (SBL-O)

19.3 Why? (SBL-O)

19.3 Why Not? (SBL-O)

19.3 Why did you LIKE it? (SBL-O)

19.3 Why did you NOT LIKE it? (SBL-O)

20.1 Has this course changed your attitude about majoring (or not majoring) in computing?

20.2 Please explain.

21. Please provide any other comments you have about this course.

Coding Scheme to Describe Utterances in Online Collaboration

(Curtis & Lawson, 2001)

Behavior Categories	Codes	Description	Example
Planning	GS	Group skills: a generic code applied to expressions that encourage group activity and cohesiveness.	I know that [names] have given you good advice, but I think it's worth knowing that you need patience.
	OW	Organizing work: Planning group work; setting shared tasks and deadlines.	I just want to set a time-line for myself. Is everyone OK with that?
	IA	Initiating activities: Setting up activities such as chat sessions to discuss the progress and organization of group work.	I would like to chat on the blackboard. What about this Friday at 7.30pm SA time?
Contributing	HeG	Help giving: Responding to questions and requests from others.	To access the chat room, click on virtual chat in the blackboard; chat screen will come up; click on enter..FBG
	FBG	Feedback giving: Providing feedback on proposals from others.	I like your idea of a generic booklet and everyone contributing aspects of interesting internet services.
	RI	Exchanging resources and information to assist other group members.	With the implementation of an internet service ... there has been a major shift in the communication function in business.
	SK	Sharing knowledge: Sharing existing knowledge and information with others.	I think we also need to give thought to the following. 1. The issues of quality/efficiency in teaching and learning.
	Ch	Challenging others: Challenging the contributions of other members and seeking to engage in debate.	No examples - behavior not identified in the text.
	Ex	Explaining or elaborating: Supporting one's own position (possibly following a challenge).	No examples - behavior not identified in the text.
Seeking Input	HeS	Help seeking: Seeking assistance from others.	Does anyone know how to edit/add/append data on the student pages?
	FBS	Feedback seeking: Seeking feedback to a position advanced.	What do you think about answering the questions that ... has put forward?
	Ef	Advocating effort: Urging others to contribute to the group effort.	Haven't heard from you for a while. Are you still with us?
Reflection / Monitor	ME	Monitoring group effort: Comments about the group's processes and achievements.	I believe the overall contribution and collaboration of working as a group requires an increase within itself as part of our learning.
	RM	Reflecting on medium: Comments about the effectiveness of the medium in supporting group activities.	The email for the discussion group seems to work OK for me. You know it has gone through because you actually receive your email back almost straight away if it has worked.
Social Interaction	SI	Social interaction: Conversation about social matters that are unrelated to the group task. This activity helps to 'break the ice'.	Regarding chat - my weekend is pretty hectic - I have my family flying in from Greece ... so the Greek festivities will be in full swing.

Figure B1. Coding Scheme to Describe Utterances in Online Collaboration

Appendix C. Assignments and Inspection Forms

Assignment 6 - Calculating Body Mass Index

This assignment will help the student by:

Reading user input

Using either the Scanner class or the BufferedReader class

Validating user input

Using try/catch statements

Using "if" statements

Converting strings to numbers (if using the BufferedReader)

Students will also practice

Performing calculations

Operator precedence

Formatting output using decimal formatting

BMI Calculation Instructions:

You will write a JAVA program that will calculate the BMI for an individual.
You will require the following information from the user:

height in inches

weight in pounds

You will then calculate the BMI with the following formula:

BMI = (703 * weight in pounds)/(height in inches)^2

When you read the information from the user you should check that:

The user enters numeric input in both instances (for the height and for the weight)

The height is not less than 48 inches

The height is not more than 80 inches

The weight is not less than 75 lbs

The weight is not more than 300 lbs

If any portion of the requested input is invalid, you should display a message to the user specifically addressing the issue and you will terminate the program.
If the input is correct, you should perform the calculations and display the results in a professional and well-formatted way.
Generally a BMI above 25 is considered high, that is, likely to indicate that an individual is overweight. For your output, if the resulting BMI is high you should address the issue in your output.

Add comments as needed. Here is some sample interaction/output:

```
  Please enter your height in inches: 200
  sorry, you have enter a height that is not valid
-----
  Please enter your height in inches: 62
  Please enter your weight in pounds: x
  sorry, you did not enter a number in pounds
-----
  Please enter your height in inches: 62
  Please enter your weight in pounds: 130
  your BMI is 23.8, it is considered good.
-----
  Please enter your height in inches: 70
  Please enter your weight in pounds: 247
  your BMI is 35.4, it is considered high.
-----
```

Output messages may be different than the ones depicted here, this is just an example.

Be sure to follow the java coding standards

The program name should be BMI .java if you use any other name you will get points deducted.

Attach the file to an e-mail as we learned in class. Submit only the .java file. Do not submit the .class file. I will generate that one.

Your e-mail should have the following subject: A6-BMI/lastname (lastname should be substituted for your lastname). Not following this standard will result in a deduction of points.

Send the file to ics111_homework@hawaii.edu account.
If your program doesn't work in UH Unix you will get NO credit.
If you have any questions please e-mail the instructor.

Assignment 8 - Slot Machine Applet

This assignment will help the student by:

Implementing interactive applet components within a fully functional program.

Using Action Listeners.

The Student will also practice the following skills

Posting applets on the web

Using Random numbers (import statements)

Implementing "if" statements

Building Boolean expressions

Slot Machine Instructions:

Create an applet that will simulate a slot machine. The applet's name should be "SlotMachine.java". You are free to design your code as you wish as long as you meet the requirements cited below. As usual, you should use the ICS 111 Java coding standard for this project.

Specific Requirements

When the applet loads, it will show 3 big buttons in the middle of the screen(the buttons should be white). These big buttons do not need a caption, so you can make the caption a blank string. Keep in mind that these 3 buttons are supposed to be big since they are representing a slot machine.

The applet should have a TextField which will display the number of times played. This TextField should be accompanied by a label "games played". This TextField should not be editable by the user.

The applet should have a TextField which will display the number of games won. This TextField should be accompanied by a label "games won". This TextField should not be editable by the user.

The applet should have a button labeled "play" which will randomly give color to the 3 buttons. If the three buttons' colors are the same the count of games won will increase.

The applet should have a button labeled "reset" which will reset the number of games played, the games won back to zero and the buttons will be white again.

There should be 5 different colors that may be randomly selected to color the three big slot machine buttons.

Here is a sample GUI, feel free to modify it, as long as it keeps the same Slot Machine functionality.

Slot Machine GUI:

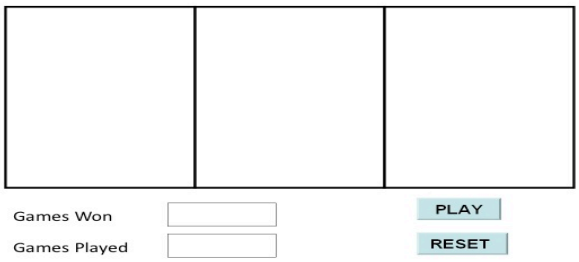


Figure C1. Slot Machine GUI

A little reminder about random number generation:

You will need to import java.util.Random;
You will then create an object of class Random
Random r = new Random( );
You will then generate a random number

```
int i = r.nextInt( ); // this will give you a huge number
int i = r.nextInt( n ); // this will give you a number between 0 and n-1
```

Be sure to follow the java coding standards

The program name should be SlotMachine .java if you use any other name you will get points deducted.

Attach the file to an e-mail as we learned in class. Submit only the .java file. Do not submit the .class file. I will generate that one.

Your e-mail should have the following subject: A8-SlotMachine/lastname (lastname should be substituted for your lastname) Not following this standard will result in a deduction of points.

Submit also the complete URL (complete, beginning with http://) in the body of the e-mail, so I can visit your webpage and see your applet running.

Send the file to ics111-homework@hawaii.edu account.

If your program doesn't work in UH Unix or doesn't display on the web you will get NO credit.

If you have any questions please e-mail the instructor

Assignment 12 - CD Store

A driver class for your CD object and CDException

This assignment will help students:

Create driver classe for custom objects.

Use a driver class to read user input and create objects

Create an arraylist of custom objects (CDs)

Practice

communicating several classes

creating methods

use typecasting

Using arrayLists

Reading user input

CDStore Instructions:

You have written a JAVA program that creates a CD object with four instance variables, along with a CDException.java class that helps you validate this input. So far you have been using these classes with my driver classes. Now it is time for you to create a real driver class that will implement an ArrayList of CD objects.

Here is a list of requirements for your LAST program

Read the artist name from the user

Read the album name from the user

Read the number of items in stock

Read the price of the album

Your program should continue to read the CD information from the user until all the entries from the user for all the fields are blank.

Lastly, your program will then print the list all the "valid" CDs entered by the user.

Sample Input/Output INPUT

Album: The Resistance

Artist: Muse

Price: 11.99

In stock: 20

==========

Album: The Live at the Acropolis

Artist: Yanni

Price: 11.99

In stock: ABC

==========

Album: Black Holes and Revelations

Artist: Muse

Price: -2

In stock: 2

==========

Album: The Fame Monster

Artist: Lady Gaga

Price: 14.99

In stock: 12

==========

Album:

Artist:

Price:

In stock:

==========

The OUTPUT

Album: The Resistance

Artist: Muse
Price: 11.99
Currently 20 in stock
==========
Album: The Fame Monster
Artist: Lady Gaga
Price: 14.99
Currently 12 in stock
==========

As you can see the CDs with invalid entries were not added to the ArrayList, therefore they were not printed. You SHOULD use an ArrayList and not print valid input directly to the screen. You will resubmit your modified CD.java along your CDException.java along with your newly created CDStore.java. If you use any other names you will get points deducted.

Attach the files to an e-mail as we learned in class.
Add your name as a comment to your code
Submit all three .java files. Do not submit the .class files.
Your e-mail should have the following subject: A12-CDStore/lastname (lastname should be substituted for your lastname) Not following this standard will result in a deduction of points.
Send the file to ics111-homework@hawaii.edu account.
If your program doesn't work in UH Unix you will get NO credit.

The following are the instructions for the CD and the CDException classes that go along with this final assignment:

Assignment 10 - CD Object with Exception

This assignment will help students:

Create their own exceptions

Use set methods as a way to validate input

Create their own exceptions

Practice

communicating several classes

creating methods

CD Object with Exception Instructions:

You have written a JAVA program that creates a CD object with four instance variables. You will modify this CD.java program to interact with an Exception that you will create. This will be the CDExcection.java program. Your modifications to the CD.java program will allow the CDStore2.java program to create CD objects without ever creating invalid objects The CDException will prevent the following errors from happening:

Artist name should not be blank or contain blanks only

Album name should not be blank or contain blanks only

CD price should not be negative (zero is ok, it means that is free)

CDs in stock should not be negative (zero is ok, it means they are out of stock)

Your code should produce the following output:

Album: The Resistance

Artist: Muse

Price: 11.99

Currently 20 in stock

==========

Album: The Live at the Acropolis

Artist: Yanni

Price: 11.99

Not in stock

==========

Album: Black Holes and Revelations

Artist: Muse

Price: 10.99

Currently 15 in stock

==========

CD price cannot be negative

Cannot create CD

==========

Album: The Fame Monster

Artist: Lady Gaga

Price: 14.99

Currently 44 in stock

==========

Artist name contains no characters,

Cannot create CD

==========

Album name contains no characters,

Cannot create CD

==========

For this you will use the java code listed at the bottom of this page.
You will resubmit your modified CD.java along your newly created CDException.java. If you use any other names you will get points deducted.

Attach the files to an e-mail as we learned in class.
Only submit the CD.java and the CDException.java files. Do not submit the .class file.
Do not submit the CDStore2.java file.

Your e-mail should have the following subject: A10-CDException/lastname (lastname should be substituted for your lastname) Not following this standard will result in a deduction of points.

Send the file to ics111-homework@hawaii.edu account.

If your program doesn't work in UH Unix you will get NO credit. Your program should run with the following class that I am providing for you:

```
//author: Blanca Polo
   //driver class for CD.java and CDException.java

   public class CDStore2{

     public static void main (String[ ] arg) throws Exception{

     try{
          CD cd1 = new CD("Muse", "The Resistance", 11.99, 20);
          System.out.println(cd1.toString( ));
          System.out.println("==========================");
       }
      catch(CDException cde){
          System.out.println(cde.getMessage( ));
          System.out.println("==========================");
      }

     try{
          CD cd2 = new CD("Yanni", "The Live at the Acropolis", 11.99, 0);
          System.out.println(cd2.toString( ));
          System.out.println("==========================");
       }
      catch(CDException cde){
          System.out.println(cde.getMessage( ));
          System.out.println("==========================");
      }



     try{
          CD cd3 = new CD("Muse", "Black Holes and Revelations", 10.99, 15);
          System.out.println(cd3.toString( ));
```

```
        System.out.println("==========================");
      }
     catch(CDException cde){
        System.out.println(cde.getMessage( ));
        System.out.println("==========================");
     }

try{
        CD cd4 = new CD("Paul Mc Cartney", "All the Best", -0.99, 12);
        System.out.println(cd4.toString( ));
        System.out.println("==========================");
      }
     catch(CDException cde){
        System.out.println(cde.getMessage( ));
        System.out.println("==========================");
     }

try{
        CD cd5 = new CD("Lady Gaga", "The Fame Monster", 14.99, 44);
        System.out.println(cd5.toString( ));
        System.out.println("==========================");
      }
     catch(CDException cde){
        System.out.println(cde.getMessage( ));
        System.out.println("==========================");
     }




try{
        CD cd6 = new CD("", "California Gurls", 1.29, 4);
```

```
      System.out.println(cd6.toString( ));
      System.out.println("==========================");
    }
  catch(CDException cde){
      System.out.println(cde.getMessage( ));
      System.out.println("==========================");
  }



try{
      CD cd7 = new CD("Pitbull", "", 11.99, 12);
      System.out.println(cd7.toString( ));
      System.out.println("==========================");
    }
  catch(CDException cde){
      System.out.println(cde.getMessage( ));
      System.out.println("==========================");
  }

}//main method ends
```

Inspection Forms

GROUP INSPECTION

Code author: _____

Recorder: _____

Other reviewer(s): _____

Instructions: The Author should briefly present his or her code and point out any noteworthy elements or known problems. The Reviewers should then report and discuss their individual findings and come to a group consensus on the following questions. The Recorder will record the group's conclusions here.

SYNTAX

1. Compiles. Does the code compile?

YES – Compiles

NO – Few/minor errors

NO – Many/major errors

Comments:

SEMANTICS

2. Correctness. Is the program complete and correct?

YES – Complete; no errors

NO – Complete; single error found

NO – Incomplete and/or multiple errors found

Comments:

3. Runtime errors. Does the program crash at any time?

NO – No errors

YES – Single error found

YES – Multiple errors found

Comments:

4. Elegance. Did you notice any particularly good, clear, or simple sections of code? In contrast, was there any unnecessary, repeated, or redundant code?

GOOD – Elegant and robust

OK – Nothing noteworthy either way

POOR – Unnecessary or inefficient code

Comments:

PRAGMATICS

For the following statements, note your agreement according to the following scale:
5 – Strongly Agree **4** – Agree **3** – Neutral **2** – Disagree **1** – Strongly Disagree

Code Formatting:

5. The code was easy to read. **5 – 4 – 3 – 2 – 1**
6. Variables names were descriptive and relevant. **5 – 4 – 3 – 2 – 1**
7. Code was properly indented so that it was easy to match up braces. **5 – 4 – 3 – 2 – 1**
8. The comments and documentation were clear and helpful. **5 – 4 – 3 – 2 – 1**

Comments:

User Interface:

7. The purpose/function of the program was clear. **5 – 4 – 3 – 2 – 1**
8. Instructions to the user were clear. **5 – 4 – 3 – 2 – 1 – NA**
9. It was easy to enter input correctly. **5 – 4 – 3 – 2 – 1 – NA**
10. Error messages were clear, helpful, and explained what was wrong. **5 – 4 – 3 – 2 – 1 – NA**
11. The program was generally easy to use. **5 – 4 – 3 – 2 – 1**

Comments:

Features:

12. The extra features (if any) were valuable additions to the program. **5 – 4 – 3 – 2 – 1 – NA**

Comments:

GENERAL NOTES & COMMENTS:

Appendix D. Session Transcripts and Learning Moments Script

Figure D1. Session 1–Team 1 Transcript

PRI	SEC	Chat	PRI	SEC	Talk
			RM		C133: and you guys see this? Floating around…
			ACK		B131: yes I can
			ACK		C133: okay
			COM	TLR	C133: my version is a little bit more complicated.
			TLR		C133: there is a lot going on.
			FBS		C133: did it work on everybody's computer?
			ACK		B131: it worked.
			RM		B131: There's a lot of background noise.
RM		C133: steel echoing? .	OW	IA	IM: okay guys allow me just to say something. You highlight a couple of lines of code and you explain what you did. And then the next few lines and explain. And so on. Okay? And now I'm leaving.
ACK		C133: okay			
RM		C133: I think when there's 2 speakers it creates an echo. .			
			EXP		C133: Hello again, ok…for my variables I have my 2 strings, the height and weight; I needed it to be for the input. And I have to convert them to doubles so that they would come out as a number. When I needed them. When I needed my calculations. [*]
			EXP		C133: my final integers are the max, the minimum and the Max my low height high height low weight and height weight. My objects for input to read the answer and convert my integers to decimals well with the decimal format.

PRI	SEC	Chat	PRI	SEC	Talk
			EXP		C133: my input for height and [*]. This converting my string into a double so it comes out as a number, so read [*]. Then weight in pounds, converting my weight so that it comes out as a number. My calculations for BMI, which I googled then I realize that my numbers were coming out wrong. [laugh]
			EXP		C133: my 4 ifs. This is my 1st one for height. To make sure that the height was between 48 and 80 inches. And I know that somewhere over here, my error code is wrong. And they don't, come out right. But, this one is for my weight to make sure that eats between 75 and 300.
			TLR	EXP	C133: if the BMI was less than 25 "your BMI is". And my BMI calculations are right here. And then if the BMI is greater than 25.
			FBS	EKE	C133: I know that this part is the messed up part. I don't know, I would like to hear from other people.
TLR		C133: I hope that's not confusing!	EXP	TLR	A100: hey, this is A. I think that what you did everything in a nested if, if else, else and then catch at the end it looks like it worked. I didn't even think of doing it that way.
			FBS		C133: does that work for you B? Is that confuse?...Or I don't know, how yours compare to mine?
			FBG	CHQ	B131: it should be a nested if-else. If you really look at your program it just failed through the 1st check you know, when you look for height. Yes the height was invalid and it didn't matter, you still displayed your BMI. I know the formula is wrong and that's irrelevant. But, it didn't stop at that point, and it should have stopped. So a nested if-else could actually be your best solution.
EXP	TLR	C133: it just seemed convenient for me.			

PRI	SEC	Chat	PRI	SEC	Talk
RCW		C133: I e-mailed Prof. Polo about that! She never got b to me = (			
HES		C133: how do I do that?			
ACK		A100:?			
RM		C133: can you hear B, A?			
RM		B131: having trouble hearing me?			
HEG		A100: I used "system.exit ( );" to exit and println" each time for invalid input.			
FBS		C133: and that gave you the error after each input?			
			FBG	COM	A100: this is A again. Yeah, I went about this a little bit differently. I made a, instead of putting it all in one try-catch I would do, let me see, one, two, I guess I did a try-catch for each of the height and weight. So I did try, if it was [*] and then I use the scanner, the method, the object… and then to catch the errors… the user input to like an incorrect or are of bounds error and then each time if it was an error I just exited and I printed " you just entered an invalid height or weight". So I tried for height and if it was a good height then I went on to try and catch for weight and then if I got it good input for that then I did the calculation for BMI and I used like a if-else at the end for the considered good or considered high.

PRI	SEC	Chat	PRI	SEC	Talk
NTA		IM: I'm here in case you have any questions.			
			FBS		A100: I don't know if that makes sense, because you guys are not looking at it. Or maybe you guys are picking up what I'm saying.
			FBG	COM	B131: this is B I did the same thing. Yeah, you got to isolate it. You can't get to the finite detail like if they are entering the height of wrong [*] invalid character, you can't give them the right message. Then check out the weight after that try-catch, so you introduce another try catch to see if they are entering the pounds correctly, if they answer the wrong thing, so that you can present them with the proper message, so the user know what's going, yeah, on per se.
AGR		C133: that makes sense.			
GS	NTA	IM: I will be b in another 10 min. you guys keep it up :-)			
RM	FBS	C133: you guys can still see my program right?			
ACK		A100: yup			
ACK		B131: yes			
ACK		C133: okay, with it just look on/not work if it did…			
			HYP	TLR	A100: I think this would work still.
			RM	ADE	A100: Hey B, if you are not going to talk, you may want to kill the mic.
TLR		C133: nevermind. I			

PRI	SEC	Chat	PRI	SEC	Talk
		have no idea what I just did			
RM	ADE	A100: B, can you turn your mic off if you're not talking?			
RM		A100: it makes an awful echo for me.			
ACK		A100: thanks!			
			COM	HYP	A100: I guess I will throw these one to B. I tried it ran C's code compiled, and I ran it and I try to throw letters in the different strings added to get to error but it didn't, each time it completes the program it makes you enter in, if you can enter, I guess if you enter an invalid height, and then he will get the invalid weight but it will still tells you the error message. This is actually less lines of code than what I wrote. So I cannot consider that this was actually a little bit of… I guess it was a good thing. If you can do the same results with less lines I think it's a good thing.

PRI	SEC	Chat	PRI	SEC	Talk
			COM		B131: well… yeah let lines of code, but the code wasn't elegant in a sense that it not solved for problem. So, it was nice and tight but you know, when you add all this to try the code does expand. My code in size is about 8 kB but I have a lot of comments in it and she doesn't have any comments so, you know, with more comments sometimes it makes your code group a little bit more fluffy I guess in the sense, you know, sometimes it's easier to read sometimes it's hard to read. The thing that bothered me is just basically how she handles her exceptions in the beginning it should have been a try then and nested try-catch again just to catch your error on the input. Yeah, it just falls through like you said so maybe, you know [*] so to see exactly where she went wrong.
EXP		C133: the only time it crashes for me is when you enter a wrong weight.			
NTA		C133: don't mind me			
EXP		C133: sorry, I was trying to put it b to normal	OW		A100: there you want to go at it line by line? I think you were saying so you want to look at it line by line?
			IA		B131: let's look at it section by section.
			RM	IA	A100: okay, I have it. I think we are all looking at it right now. I can see it in the application sharing window in the Elluminate. But I have it open, I just copied it into notepad. Real quick to Do, we go through piece by piece.
			EXP		B131: yeah I opened C's code in Java, actually in UNIX [*] you can do that.
			ACK		A100: okay she has, ah…

PRI	SEC	Chat	PRI	SEC	Talk
			HYP		A100: I guess variables 1st. I mean, she imported Java.text and Java.io
			COM	FBS	A100: at 1st I was wondering why she used strings to grab the height and weight as user input maybe I'm just lazy, [laugh]I like the scan methods because you don't have to use the buffer reader and input stream reader I think it's less just have to write one line, you don't have to declare as many variables, so I was looking at that and I was wondering maybe I'm missing something? Maybe some benefit from taking user input as strings and then convert them into double. Was there something going on around there? And does that benefit in any way? I wonder.
			TLR		C133: I do it because it is a force of habit. I don't think… I rarely use the scanner. (Mumble)
			EXP	RCW	B131: I guess for me the scanner, when Blanca was talking about it in one of our lessons you know basically a person could eliminate input could add at more than one line of input and then delimit it with commas I didn't know it was going to introduce more errors into my program or not, so I'm more used to taking in one input that's entered and that's it. That's the reason why I used the buffer reader method instead of scanner.
			FBS		A100: you say that you chose that because it's safer? If you read a buffer generates potentially less errors?
			EXP	FBG	B131: I guess I just want to accept one input, you know? Let's say if somebody types in the keyboard and accidentally hits a comma it just takes another kind of input. I don't know how the scanner takes the second input but to be on the safe side I'm not to create problems for my programming I guess I just stayed with the buffer reader for convenience.

PRI	SEC	Chat	PRI	SEC	Talk
			TLR	HYP	A100: okay… I wasn't factoring in commas, I'm trying to think how I did… I guess I [*] I don't know how would I've dealt with it if somebody would you have put in a comma. I guess, I would only would have taken a double but yeah I don't know. So you're saying that the scanner, if you're using the scanner object or method, yeah, somebody throws a comma in there that it forces it into error. I didn't even try that.
NTA		IM: I don't want to interrupt but I will be b in 5 min	EXP		B131: Well, I'm new to Java programming too, so I don't exactly understand how the scanner works because I haven't played enough with it. I just want to cut down the problems, you know, maintenance wise. I just wanted to be able to deal with an error that I did know how to handle.
ACK		A100: k		AGR	A100: yeah, that makes sense, that makes sense.
ACK		C133: k			
			CHQ	FBS	B131:C,-sorry about that-, C going back to your code ah, on the final, you have all those constants. I remember Blanca saying that all those constant names should be all capitalized. Did you just forgot or?... What's happening there?
			FBS		C133: the final ones, or my strings?
			FBG		B131: in the area where you define them.
			EXP	FBS	C133: I guess I just left them that way because it worked. Then I didn't change them because it didn't make a difference in the compiling sense of it?
			SKN	RCW	B131: It doesn't make a difference, but what Blanca was trying for us to do, there is a Java standard that she would like us to follow and I assume that the whole Java community follows that standards. I have been looking at books and it seems that that's what they do with constants.
			AGR		A100: you mean?… Oh! Ok, I see what you mean. [*]

PRI	SEC	Chat	PRI	SEC	Talk
			SKN	RCW	A100: I didn't do that either. I capitalize the 1st letter. But I didn't leave them all, but I think you're right. I remember hearing that or reading that. They are supposed to be all caps according to the Java coding standard. Good practice!
			SKN	RCW	B131: One thing that I like that Blanca teaches that is interesting, she puts like a small “i” in front of an integer variable name, she puts an s in front of the String variable, she puts a d in front of a double. In a way that’s kind of important as you are playing with numbers. If you have weird calculations and you don’t know what you are really doing and you want to know what you are really dealing with and you know what the outcome is gonna be.
			SKN	RCW	B131: Let's say that I am working with code that is 40 or 50 pages long I don't want to be flipping back and forth to figure out is these adult will or is it an integer. Or is this a string I'm dealing with? It's a little bit hard to read so I kind of like what Blanca did you know putting the i in front of it or the letter s in front of it. I kinda know what it is, she has done it in the variable declarations, in some of it like for the doubles. But like for weight it doesn't have an S. In front of it. for integers it doesn't have the I in front of it. It makes it harder to figure out what is a variable representing. A double an integer or a string?

PRI	SEC	Chat	PRI	SEC	Talk
			TLR	FBG	A100: yeah, I do that too. If its a string you do lowercase s. and then capital like in this case it's named height so I will put lowercase s, uppercase H and then lowercase for the rest of the spelling of the word height. And yeah, I think you are right.[*] It is pretty explanatory that it says height, but if you have a program that is three pages long you may be scratching your head thinking if height is string or a double, so yeah, I would say, that kinda helps. It doesn't compile any easier, you know?
			RCW		A100: You don't have to do it but if it is one of the standards I guess it probably helps your grade. I don't know if Blanca grades on that or not.
			RCW		B131: I think that subconsciously, if you do the things that she likes she will probably be easier on you if you make the mistake. I would have gotten all perfect in this program, it is just that when I named my program I used capital "B" then "mi", it was supposed to be all capitals and that's why she marked me off. Nothing on my code, but just that.
			SKN		A100: this works though! What do you think about… I guess I'm skipping around here, but I'm looking at it… you go try grab the height, and then is the buffered reader to read the height. Read line then [*] it parses the double for height. I mean, with the weight, it does the same thing. And then it calculates.
			TLR		A100: I was wondering, like I was thinking mmm is calculating in the middle of the try-catch. I don't know, I don't think that matters. I did it at the end. Ammm.
			HYP	FBG	A100:The only suggestion I will say here is that maybe it would run faster if you would have like a more complex program. If you didn't have to calculate …

PRI	SEC	Chat	PRI	SEC	Talk
			TLR		A100: But if you have a catch situation you don't need to catch you can just exit through the error message the invalid height or weight. So I don't know if it needs to be there. I don't know, I'm trying to think out loud if that is going to change the structure of the program. I kind of like the structure of the program it's better than mine. I mean, it works.
			FBS		A100: B, were you able to make it err? Not in the sense that you will I have an invalid statement but to actually crash it? Were you able to do that? I'm wondering.
			FBG	COM	B131: no, hard to crash something with a buffer reader. Because basically, if you go with the invalid input is really not a problem at that point. If you guys enter numbers, no problem because takes it out at that point. But, I would have put another try-I cannot scroll the program that you guys are now scrolling right now but after she set the input for the height in inches that's what I would have… another else and try then just to enter the pounds. She did calculate the BMI right after the 2 inputs…. It looks okay, I guess, but, I guess I'm old-school in the sense that I would take it with the flow, [*] exactly like how the user would use it, then I would go on with the calculations, she doesn't do any of that. She does some with the invalid input of height and weight but she does the calculations for the BMI and then she does the filtering for the [*]
			TLR	EXP	B131: weight and height, and that's another edit issue. So, I like to keep all my edits up front. So if somebody else uses my code and the he asks me. Oh, ok and here is all the edits are. And here is where all the calcs are, [*] and at the end there is a output. Test the output. Then you can read it, it's an easier flow I guess. That's my way of looking at things.

PRI	SEC	Chat	PRI	SEC	Talk
			TLR	HYP	C133: when I move my BMI calculations when I was compiling it, when I was making it, it will not work when I moved, the way that I just moved it. I guess if I had moved it higher maybe, but when I moved it here it will not do the math for me. For reason [*]. I wonder why. I guess, if I take it out. Hold on…
			EXP		C133: yeah, then it doesn't even give me anything. That's why I left it there.
			TLR	HYP	A100: yeah, it works up in the top, so? I don't know why it wouldn't work down lower. I guess it's part of the try and then I don't know why it will only work at the top if you figure it would work if you put it after the catch, well, I don't know about that. Yeah you will think it will work lower. I just, but..
			HYP	TLR	A100: the if and then applies like the OR statement I didn't think to do that and that actually condenses the code a little bit. It think you can, well, oh no you still need it to catch errors, but yeah, I have a bunch of… like I said I have two try-catch statements so there is much more lines. But if I could have done this IF OR statement or use that OR operator because it's an inclusive OR, it would work but, I didn't think to do that.
EXP		C133: I'd rather have more lines than get an error with the wrong height :-)			
			FBS		A100: are you getting an error with the wrong height? I don't understand.
ACK		C133: nope			

PRI	SEC	Chat	PRI	SEC	Talk
NTA		IM: I'm here. Do you have any questions otherwise I will disappear again :-)	FBG	CHQ	B131: C, did you ever thought about moving the calculation for BMI after the else that you have there?
			FBS		C133: when it is after the else it calculates and then it gives me zero. Can you see my program?
			ACK		B131: yeah, I see your program
ACK		IM: yes	ME		C133: okay, hold on (singing)
ACK		A100: Yep	HYP		C133: like if I put it… There [*]
NTA		IM: okay guys I will be back.			
			FBS		C133: a small height, or I should put in a good height? zero
			EKE	FBS	C133: I just realized that's when my BMI calculation moves I get an error for my height. That's weird. But if it's the right weight I don't get an error for my height and I get one for my weight, [laugh]
			FBG		B131: if you put your BMI after the catch, when you come to the part where you say if it's out of range yeah it's not gonna work because the BMI is calculated after-the-fact yeah?
NTA		C133: experimenting			
			FBS		A100: do you think that has something to do with the type of catch that she is using?
			COM	TLR	A100: I just realized that I used, it's a different catch statement. Like mine says catch parents input mismatch exception space IME –close parents- and then it looks like she is using a catch number format exception NFE, so now I am a little confused because that's a different catch than I used.
			NTA		B131: sorry, I'll be right back , I am trying to compile C's code and see what happens I'll be back .
ACK		A100: k			

PRI	SEC	Chat	PRI	SEC	Talk
			FBG		B131: C when I compile your code [*]putting the formula after the else it does work I have no problem with it. It does calculate the BMI. I don't come out with zero or anything
			FBS		C133: which else did you put it after? [*]
			FBG		B131: I put it after the else right before you actually display the results for your BMI. That's where I put it, so I little bit higher.
			FBG		B131: yeah, that's where I put it and when I compile it, I get the results
TLR		C133: so now I just need to crash immediately after I've done something wrong.			
			CHQ	EXP	B131: no I actually you still have some more work to do. You know, where you start setting up the ranges for your BMI where you are saying if it's good the bad and so forth, there are supposed to be four categories, obese, overweight, normal, and underweight. And you are not factoring that in. You are kind of clumping the overweight and obese in one category and the normal and underweight in another category. I think she wanted it all broken up.
RCW		C133: I only saw to on the assignment page. Just if over 25	TLR		B131: I may be wrong, I didn't look at the assignment sheet. I don’t have it in front of me, maybe you may be right. I guess I did a little bit more work than normal.

PRI	SEC	Chat	PRI	SEC	Talk
			RCW		A100: I just pulled the assignment sheet open. And it says when you get the information from the user you should check that the user enters numeric input in both instances for the hight and for the weight and then checking that the height is not less than 48 inches and the height is not more than 80 inches and then the next bullet is the weight is not less than 75 pounds. And then the final bullet that the weight is not more than 300 pounds. And then she gives an example for output but basically I think it's supposed to say [*]…… Yeah BMI above 25 is considered high. And the other BMI it just defaults to okay under 25, below or including 25 or above 25 is high.
			TLR		B131: I guess I just worked too hard then.
			FBS		A100: what did you do again? It had different parameters? Or more strict parameters?
			SKN		B131: what I did is that I just broke it up a little bit more. I'm kind of more familiar with BMI, I work in a hospital, that's why. Yeah there there's categories of underweight, normal, overweight and obese and I decided to break out all the categories. So, It showed I guess, a little bit different. I know that you saw my code, yeah? So you kind of saw that's when I, before all the input I have an explanation about all that I was doing, basically, so that's how I coded.
			TLR	FBG	A100: no, I didn't see your code. I thought, maybe I just missed an e-mail? I don't know, I was thinking about that when I was going over the review sheet.

PRI	SEC	Chat	PRI	SEC	Talk
AGR		C133: she only sent mine	AGR	SI	A100: yeah, C is saying that she only sent hers. I didn't see yours. So she is kind of in the hot seat here. [*]
			EXP		B131: I sent mine as an attachment on the next day that I received that e-mail. She sent me 2 e-mails and I attached to the 1st one that I saw.
			NTA	RM	A100: okay, I'm gonna go back into Pine real quick and make sure that… Let me see if I can open it up real quick.
SI		IM: hi guys	FBS		IM: you sound real quiet. I'm here to ask you guys have any questions.
			SI		B131: can we pay you for a higher grade?
			GS		IM: [laughing][*]
			TLR		B131: I agree everybody has a different way of solving a problem. Coding is just coding, but solving a problem, depending on how the person thinks.
			AGR		IM: That's right, it shows your thinking process. And it shows your personality also. [*]
			FBS		A100: was there a difference? It looks like I used a different exception in the catch.… Exception code, I don't know another way to say it but is there a difference between this NFE and the IME, like input mismatch exception and number format exception?
			FBG		IM: it is, when you have the number format exception is because you are using the buffered reader and it usually happens when you are trying to convert a string that you read from the user into a number that's when it happens. It doesn't

PRI	SEC	Chat	PRI	SEC	Talk
					happen at the reading time. It happens at convention time. You see, the buffer reader reads a string, and then you parse it into a number, right? So in the parsing time is when that exception happens. However, the input mismatch exception happens at input time because the buffer in this case the scanner, is actually expecting a number and when you put it in and it's not a number, it's something else, it doesn't match the preconceived idea of Java about how a number looks like, so he throws exception right away. And the difference are, that for each one of them, it's a different class that throws the exception, to begin with, so it is also in a different p age you know?
			TLR		A100: well thanks, I just have to learn more about the ups and downs of using scanner or buffer for the future
			NTA		IM: Ok guys I'm gonna let you finish. I will be back in another 10 minutes, ok? It's already 8 at night. I guess you are having a lot of fun! Ok, I will be back in 10.
			NTA		A100: hey B! I got your program open, your code open.
			FBS		A100: you imported Java.util also, I guess that's the one you have to use for the scanner. Did you use a scanner too? [*]
			COM	FBS	A100: You imported more than I did, and also more than C did. And I'm wondering (I don't see it right here) why did you did that?
			EXP	FBG	B131: Too lazy to edit my code, I just copied all the includes that's all I did.
			ACK		A100: gotcha
			HYP	COM	B131: I presume that my code probably looks a lot like yours in sense of logic.

PRI	SEC	Chat	PRI	SEC	Talk
			AGR	COM	A100: I think so, man! You have a lot of else statements. I feel like I didn't put enough. Let me look it up for a second more. Let me see. Okay, I think I see what's going on. You just made it a lot more granular and you are giving the user a lot more data and I guess that's what you're coming from, as you said. If you work in a hospital, you definitely, you put more categories. But, it is a good chance to practice the if and else statements.
			TLR		B131: so you kind of can see how I nested that if-else, that's one thing, you know, that you do more and you get better at it.
			TLR	RCW	B131: I don't know if C will ever get my evaluation, but in my evaluation I said, I don't know if Blanca would agree with me, but basically I look at coding as telling a story of how to solve a problem and if you don't have it in logic and your mind is all over the place, your code tends to be all over the place so I try to tell a story, it is like baking a cake. There is a step-by-step logic to it, you get the ingredients, you mix them, you put it in the oven and so forth. That's the way I intend to code.
			EMS		B131: C, don't feel bad, I'm like that in English. That's why I can't get out of English class.
TLR		C133: my brain goes in 1 million different directions.			
TLR		C133: all the time.			
TLR		C133: I have a lot of programs			

PRI	SEC	Chat	PRI	SEC	Talk
TLR		C133: they only get worse			
TLR	SI	C133: :-)			
			TLR	COM	A100: I guess I'm the only one, or the only one in the group, and the only person that used scanner. B, didn't even take a look at the scanner. Given you think oh, that looks like a pretty good way to get input from the user.
			TLR	SI	B131: well, like Blanca said, when she talked about the scanner, and she mentioned about the buffer and all that other stuff, I wasn't comfortable with that because, just to give you a little bit of background on me, you're talking to a guy that it's really, really into computers from a long time ago. When Blanca was talking about those cards, that's when I actually did some programming. COBOL, RPG, some basic. All languages are different about how they handle things. So, when I deal with inputs and things like that, I want to do it at the simplest, basic, possible way. Because, when you get too complicated, you do more things to your program that you may not need, it makes it harder to troubleshoot a program.
			SI		B131: I'll give you a “for instance” I used to work with a program that was 250 pages long. With about 66 lines on each page. Can you imagine trying to edit a program like that, unless it was structured well and have logic. You didn't want to touch that program.
			AGR		A100: yeah, that makes sense, because, I guess yeah it just reduces the opportunity for error. Trying to avoid. Every chance you can.
			TLR		A100: I guess I was just trying to, I wouldn't say code around it, but just it just looked like a good way of getting the input.

PRI	SEC	Chat	PRI	SEC	Talk
			SKN		B131: I agree, but only problem is the scanner came in I think, in Java 5, and I don't know what people are running in terms of Java, I was always a programmer and I had to code I make sure that my program was available to most people the are using Java and since buffered reader was there almost from the start, I know that it would be like a common denominator, because if you're not running Java 5, the scanner wouldn't work for me.
SDR		C133: this looks gross			
SI		C133: don't look at it!			
EMS	SI	A100: haha. You're making lots of changes. I hope this is helping you.			
			CHQ		B131: C, I have one question what if your BMI was exactly 25? What will it print out?
IA		C133: let's see	HEG		B131: don't run the code. Don't compile it. Just look at the two statements and see if you can make, just one simple change to make it work.
			SDR		A100: oh yeah! I didn't catch that! That's a good catch!
			EMS		B131: [*]like me, you always get in trouble.
FBS		C133: sorry, you said don't compile it?			
			AGR		A100: I see what's going on, is that OR statement. So, it's true both ways. If it is equal to 25.
			RID	HEG	B131: C, what I would change is, on your statement, where it says BMI is less than 25, put that = sign in front of that <.

PRI	SEC	Chat	PRI	SEC	Talk
					According to what I see in the assignment sheet if it is greater than 25, then it's considered bad. So you can say that 25 is okay.
			HEG		B131: that should catch your missing 25.
			FBS		A100: do you have to put that equals after the >?
			AGR	FBG	B131: yeah, you're right, correct it.
EMS		A100:*clap*			
			CHQ	RCW	B131: C, one thing I would change is the way your catch statement is. You say you say it's not a valid number. I don't know if Blanca would like that kind of message. If you write something out little bit more descriptive it will be a little bit better. For instance, [*] instead of saying not a valid number, you can say-for weight, please input the range is between 75 and 300 and for height between 48 and 80 and that might be little bit better message. And that will tell the user what they should input, the next time, when they actually do your program.
			FBG	CHQ	B131: you shouldn't write that the number is out of range because you kind of address that a little bit earlier. But you should in that one, because you buffered reader [*] number. So that's where you should write to say, you know invalid input please input for weight between 75 and 300 for I'm height between 48 and 80 and then please enter for weight between 75 and 300.
			COM	FBG	A100: that's what I didn't do. I didn't want to type it out every single time. Because I added it in a couple of if statements. So I just declare the string and my string was a –lets see here- string called S invalid height and then he says sorry you did enter an invalid height please run program again and enter a height between 43 and 80. And then I did the same thing with weight. Then if it

PRI	SEC	Chat	PRI	SEC	Talk
					was an invalid height it would print that line. And then the same thing for weight. It would just print the same line for any invalid height and then also for any invalid weight.
			FBS		B131: C, is that confusing?
ACK		C133: just a little	EXP		B131: what we're saying is you know, when you do the try, the try will kick you out if they put in an invalid item for instance, when you actually do the conversion from double parse double to actually [*] when the program is gonna kick out. It's when it's trying to get the string and convert it into a number that's why, you get the number error that you find at the bottom. What we're trying to say yes, put that information right after the catch that will help you keep it adjusted even for invalid height or invalid weight. And that will give the user an idea before to enter in the next time when they run your program.
			EXP		B131: so, the other thing you got correct, in your code, you are using 703 in the calculations of the BMI. Blanca will call that a magic number. So what I did, is, in my variables I declare that as a constant and I made a comment saying this is a BMI constant used for my BMI calculations. That's what I would do. So you don't get in trouble. Now, another area where you have a magic number if when you do the comparison to 25. So what I would do is to define 25 as something as the normal BMI so you can say if it is below normal BMI you can say it's good if you. If it's above the normal BMI then it's bad you know, and so on.
			EMS	FBS	B131: was that confusing again?
			EXP	RCW	C133: no, but because she didn't count off for it. I don't know why, because normally she would have. I'm going to

PRI	SEC	Chat	PRI	SEC	Talk
					leave it as is. Normally when I do those magic numbers she counts me off for it but the only thing I got counted off for this one was my error messages.
			EXP		C133: So, so that I don't confuse myself or to add any extra variables, I'm gonna leave that one alone.
			CHQ	RCW	B131: if I were you, I would change it, for the reason that she may have missed it in the 1st time, if she catches it is in the 2nd pass then you'll be in trouble because then you will get marked off when you could've corrected.
			SDR	TLR	A100: I guess I gotta change mine too then, because I just put the 703 number in there. I figured, the calculation for BMI would never change so it's not really a magic number. Just because it's always going to be, it's like π r2 for the circumference of a circle, I mean, the number would never change.
			AGR	RCW	B131: I agree with you guys in that, but check this for instance, in my midterm you know when you divide something by 2, to just assume divide the number in half. She actually marked me off and called it a magic number.
			AGR	RCW	A100: yes I got dinged 1 point for that too. And in one of the assignments, I think it was… I can't remember which one it was… the dice number, I put the number 6, as a 6 sided die, just figured that you will never ever changer dies, most games have 6 sided dice but then you can have like a multiple sided die like a 12 sided die or something like that, so I guess it can be considered a magic number. But yeah I know, you can always declare it in a variable. Using the number as a variable.
NTA		IM: I am here			

PRI	SEC	Chat	PRI	SEC	Talk
OW		IM: you have been in this session for about 90 min. We must wrap it up in the next 10 min.			
			FBS		B131: Ms. Polo! I have a question! In this BMI program, did you ever looked for magic numbers?
			FBG	FBS	IM: [*]sometimes, I get dizzy of looking at code, so some of them may have escaped me but I usually do look for magic numbers. I think, there is instances, maybe during the exam, like when you guys were looking for the random number and, you know the range, I thinking that one I was more lenient because just declaring those numbers is gonna take you like an hour to declare all that stuff [*] but I usually try to look at them, because, at this point in your life it doesn't give you a problem but, my God! Later on it will. That's wy I am picky. But yes I try to, the skip something?
			FBG		B131: no, you didn't miss anything (sarcastic)
			SI		IM: hey! That was sarcastic! [*] supposedly, because that's my job.
			OW		IM: okay guys, you have been here for an hour and 30 min. so I need you guys to wrap it up. This is supposed to go from one our 1 hour and 30 min., sometimes is less but not usually because it usually takes 1 hour at least so, it's time to wrap it up. Do you have any more questions for me or for each other? Before I continue talking…
			OW		IM: The one thing is, that I want to offer you guys, that's when you feel lost, you can come to Elluminate and check out if someone else is there you just need to click the link you clicked and you can

PRI	SEC	Chat	PRI	SEC	Talk
					meet with people I took to each other and say let's go to Elluminate.
			FBS		…so you guys need some time to say goodbye? To wrap it up, I mean?
			EXP		IM: I don't want and you guys to just stop it cold. [*]I'll give you time for you to wrap it up and then I will stop the recording and someone[+]
ACK		B131: no			
ACK		A100: nope			
ACK		C133: nope			
FBS		IM:??			
NTA		B131: C needs more input from the group			
ACK		C133: nope, I think I'm okay. I appreciate all the help!			
			OW	RCW	IM: now, guys, don't forget that you have the opportunity to resubmit your code, you can fix whatever is that you come across. Maybe you have nice cold, but you have a better idea of how to fix something please feel free to that. And remember, based on the fact that you already attended this lecture, these thing work, your resubmission is gonna be worth 30% so you still have a lot of points to make. So even if you already got full credit, you can always improve. There's always room for improvement in coding. So, go for it and resubmit the assignment, and then I'll grade again so, any questions about that?
ACK		C133: nope			
ACK		B131: no	NTA		IM: take care of yourselves, I'm stopping this recording right now…

PRI	SEC	Chat	PRI	SEC	Talk
ACK		A100: No			
SI		IM: I bet you guys are hungry. I am :-)			
SI		C133: I am this close to heatin up noodles!			
SI		A100: dinner time!			
SI		B131: no gonna watch 5-0			
SI		A100: ha			
		Recording stopped on Nov 1st , 2010 at 8:36 pm.			

Figure D2. Session 1–Team 2 Transcript

PRI	SEC	Chat	PRI	SEC	Talk
RM		M212: did it work?			
ACK		M212: yeah, it's open			
TLR	FBS	M212:What, what did I do?			
SI		M212: LOL ok sorry			
	SDR	M212:Oh wait, I can see it now	OW	IA	IM: Ok guys, I'm gonna share with you M's code. This is how it is going to work. Is you can't hear me at any time please click on the sad face. Ok? This is your 1st studio session. So, what you will be doing, if reviewing M code. So the way it's gonna work, it's that I am going to move into the next slide and in here you have the submition requirements of how I grade you. Ok?So in here it talks about the lines of code submission, etc. obviously some of these are already correct because otherwise I would have your code here with me. The name has to be exactly as I said. Following the Java coding standards. If it compiles in UNIX, is the author name there as a comment, I get you following the Java coding standards? And other issues you know. General things that may happen in any one of your assignments. In the next slide, here is where you will get into the code. Reading input, you know, [*] making sure there is no magic numbers. That the formulas are correct. That it validates. Every single little thing that you have to look for, ok? And this is the percentage that it is worth. I don't

PRI	SEC	Chat	PRI	SEC	Talk
					want you to grade it because that's my job, but what I want more than anything is your comments.
			OW	IA	IM: So, what you're going to do is that you will be clicking in one of these letters in here [*]or in these lines and you can write text. So, I want your comments in here, you can type it within the chat, I'm gonna read your chat, I'm going to check it out. [*] M will have to explain his code in chat. And you can ask questions via chat. Otherwise the first thing that I am going to do is that I will put M's code in the screen. I will go ahead and share it right now. This is the code that he has submitted and as you can see you can move around with the arrows okay?
			RM	FBS	IM: If you can see the code properly, can you please smile to me so that I know it is okay? R, M, okay. B can you see the code?
RM	FBS	IM: B can you see the code?			
ACK		B213: yes I can.	RID		IM: [*] I want you guys to go over here and review the code. So, somebody has to drive right? So I want M to actually show you the code, so he can highlight some text and type in the chat whatever is that he is doing. So, what I am going to do now, is that I am going to let M take over my desktop. Basically he is going to be in my computer and he is going to be able to work on this. So, M can you click in the desktop icon and request for control of my desktop? Click on the little screen… and once you click on it you can click and ask for remote control or something like that… Remote control of my desktop.

PRI	SEC	Chat	PRI	SEC	Talk
RM	HES	B213: where did the code go?	HEG	ME	IM: the code is here B, can you see it? Oh, hold on. Oh, some things happening… [*]Hold on one second.
			RID	RM	IM: Ok Guys, can you see my code now? Ok M? Ok, good… Ok M can you tell me, can you request.. ask me if you can [*]
			RID	RM	IM: M please ask if you can use my desktop so I can say yes and you can then take over my desktop.
			NTA		IM: okay M you have control of my desktop now. Wherever you move the mouse, its gonna go.
HES		B213: what am I doing now?	HEG		IM: no, nothing B. M is gonna show you guys his code. And he is gonna type what he is doing. [*]He's actually the one moving the mouse. Move it M. whatever he does, you see? He is taking over my desktop.
ACK		M212: okay	OW	IA	IM: so what I want you to do, explain to your classmates what you did. While you guys are going to be checking that he actually has the right things. And to fill up the form. I know it kind of sucks because you cannot talk.[*] So you can start from the beginning highlighting and explaining what you are doing in the line. Once M is done explaining his code then he's gonna retire kind of, you know, I quote that, and you guys are going to give the feedback. Only then, M will be talking no more no chatting, no defending himself, these is constructive criticism. Okay? You can give advice, you B and R. It's not bad stuff. So I'm going to leave you alone,
ACK		M212: okay			
ACK		M212: okay	NTA		IM: I will be back in a few minutes so that you guys can have some privacy. See you in a little bit.

PRI	SEC	Chat	PRI	SEC	Talk
SI		B213: hey guys			
TLR		M212: well as you can see my code isn't really good			
SI		M212: LOL			
RM		B213: sorry my MIC is not working well			
RM		R230: heck! I don't have a MIC :-)			
AGR		M212: same here			
FBG		B213: M your code is awesome			
RID	IA	IM: B, Don't talk just type.			
SI		B213: I stayed up late trying to figure this out.			
SI		B213: LOL			
SI		M212: LOL			
TLR	HES	M212: well I still don't know how to not except char			
EXP		M212: when the user types a letter it ends my code			
HES		M212: so how would u go about doing that?			
FBS		B213: so it ends when you run the program?			
FBG		B213: I tried running it last night it went through fine…			
CHQ		M212: no when the user types a letter instead of a number			
HES		R230: looking it up now, but isn't			

PRI	SEC	Chat	PRI	SEC	Talk
		there a type of utility that handles exceptions?			
HEG	SI	M212: LOL I really don't know			
HEG	FBS	B213: don't you need to declare your object?			
ME		B213: let me see the entire code hold on			
NTA		B213: MMMM I tried running it again			
SDR		B213: I see what you're saying			
ACK		M212: yah	IA	RID	IM: okay guys, can you hear me now? Okay guys who get this, this is what you're going to be doing. I'm sorry, I just want to try to explain to you what is going on so that we are all in the same page. M can you please highlight the first three lines of the code? You highlight them for me, you have control of my desktop. So, let's say you highlight the 1st 3 lines, your name, the import, and the public class BMI, the very first three. Just highlight them. There you go into the chat and you say "in these lines I have imported the scanner class because I needed to read. And I have done this because blah blah blah and that because blah blah then you highlight the next 3 and you say here I declare variables blah blah blah so you go little by little. Let's say you have a try or a catch then you explain why, you have an if statement you explain why you explain what you understood what is to validate. So just go 3 or 4 lines at the time highlight it and explain and when

PRI	SEC	Chat	PRI	SEC	Talk
					you're done then you let them talk its their turn. Are we cool with this?
ACK		M212: ah, ok			
ACK		M212:ok	NTA		IM: Okey-dokey then I will go back out and I will see you in 10 min. I just wanna give you some privacy guys so I will be right back.
RM	HES	B213: M what happened?			
HES	EXP	B213: I couldn't hear what Blanca was saying			
EXP		M212: well these 3 lines is where I declared my class			
EXP		M212: and imported that a scanner			
EXP		M212: these lines is where I declared my variables and initiated the scanner			
FBS		B213: what about decimal format?			
FBG		M212: yeah I did do that			
CHQ		R230: SSSSHHH your time will come			
SI		M212: LOL			
ACK		B213: LOL whoops			
RM	EXP	B213: I can't hear that's why			
RM	EMS	R230: don't worry there's no sound at this point.			
EXP		M212: this is where I asked the user to type in the height in inches			

PRI	SEC	Chat	PRI	SEC	Talk
EXP		M212: also it will loop until it meets the requirements of 48-80 inches			
EXP		M212: same thing done here but with weight			
EXP		M212: and here is where I calculated the BMI and displayed it.			
IA		M212: okay go for it			
ME	OW	R230: I think B just stepped out, so I'll go 1st			
ACK		M212: okay			
FBG		R230: for the most part your code looks great			
ACK		M212: TY			
TLR		R230: using the multiple scans you went around the loops that gave me migraines.			
SI		M212: LOL			
NTA		B213: I'm back.			
FBG		R230: also the control of white space was also well done			
HEG		R230: as for the text error that only thing I can think of is using a try/catch			
ACK		M212: ICIC			
FBG	TLR	R230: I never got it to work properly so I can't really help they are :P			
ACK		M212: LOL okay			

PRI	SEC	Chat	PRI	SEC	Talk
ADE	IA	R230: okay B your turn for some feedback			
FBG		B213: I liked how you kept your variables very clear			
FBG		B213: understandable			
FBG		B213: and to me the work that you put into this is far beyond my thought process			
ACK		M212: TY			
HES		B213: did we have to put comments for our code?			
HEG		M212: no just mine			
ACK		B213: oh o.k.			
TLR		M212: well if I could I would redo this code			
TLR		M212: seems incomplete to me			
FBG		B213: above all the code looks excellent and I may need to call			
FBG		B213: you for help LOL			
SI		M212: LOL			
RCW	SI	M212: how did you guys do on this assignment?			
RCW	SI	R230: terrible, mine had so many loops it didn't compile right.			
TLR	RCW	B213: honestly it was really challenging for me			

PRI	SEC	Chat	PRI	SEC	Talk
ACK		M212: icic			
TLR		B213: just knowing when and where my loops would end			
HES		M212: what loops did you guys use			
HEG	RCW	M212: at 1st I tried that if else way			
TLR		M212: but that didn't really work for me			
SI		B213: for me I learned program with-Python			
SI		B213: I'm used to using else if statements			
ACK		M212: ICiC			
			FBS		IM: Are you guys doing okay? Just smile if you do and if you have any questions type them in.
ACK		M212: yes	NTA		IM: I'll be back in another 10 min. okay?
ME	FBS	M212: well is that all?			
FBG		R230: I think we covered most of the highlights			
RM		B213: well besides not communicating through voice			
FBG		B213: it seems pretty well			
ACK		M212: yah			
RM	RCW	M212: I thought we were going to have hard time cuz of that			
ACK		B213: right			
SI		B213: this is too			

PRI	SEC	Chat	PRI	SEC	Talk
		funny			
FBS		M212: do you guys see that?			
RM	FBS	B213: what's that?			
RM	FBS	R230:?			
FBG		M212: on the comments			
FBG		B213: oh yes			
FBS		M212: so what else do you guys want to put			
RID		M212: you guys try typing in something			
HES		B213: how can I get into the screen?			
RID		M212: click the A on the left of the screen			
RM		B213: my computer if freezing up again			
RM	SI	B213: this is too much			
SI		M212: LOL			
HES		B213: M online 24 what is the square looking box?			
EMS		M212: I have no idea			
EMS		M212: LOL			
EMS		B213: oh			
EMS		B213: LOL			
SDR		M212: she's using Pine that's y			
EXP		M212: or I mean the SSH program			
ACK		B213: I C			
RM	FBS	M212: so you guys can see the			

PRI	SEC	Chat	PRI	SEC	Talk
		comments I wrote?			
RM		B213: I can't see it?			
HES		R230: I don't follow, what comments?			
SI		B213: oh by the way thanks for working around my schedule…			
RCW		M212: okay you know the slides?			
ACK		B213: K			
RCW		B213: the slides that were up earlier?			
AGR	RM	R230: on the whiteboard, yeah			
SI	RM	B213: damn I didn't see it			
HES		M212: do we all have to do our own?			
HES		M212: or we just do it on one?			
TLR		B213: by the time I logged in it was gone??			
HEG		B213: I think one on one			
SDR		B213: nevermind I know which one.			
SDR		B213: the scale points			
ACK		M212: yah			
RCW		B213: I believe she is doing this as an individual and team effort			
RCW		B213: that's why we had to be here at 10:30 AM			

PRI	SEC	Chat	PRI	SEC	Talk
SI		B213: I was 2 minutes late			
SI		M212: ha ha yes			
SI		B213: sorry			
RCW		M212: so that's what I mean we can see each other's comments			
SI		B213: I was Rushing home from work			
ME		M212: we all have to comment on it			
ACK		B213: ok			
ME		B213: can we go back and see the comment checklist?			
FBS		M212: comment checklist?			
RM		B213: never mind this damn computer			
SKN		B213: I think the main thing what's not to use magic numbers			
RCW	SKN	B213: I remember I was knocked off for magic numbers in our 2nd assignment			
AGR		R230: yup I ran into that too.			
			NTA	FBS	IM: Ok, I'm back, are you ok continuing to review this or do you want to start filling up the forms for[*]
			FBS		IM: did you check all the code? Just tell me yes, no, or do you want to continue with the code?
ACK		M212: yes	FBS		IM: so you are done with the code? Or is that yes you want to continue

PRI	SEC	Chat	PRI	SEC	Talk
					with the code.? I'm sorry.
FBG		R230: LOL finish with the code	NTA	FBS	IM: okay so I am going to stop sharing the application okay? [*]
NTA		B213: we finished	NTA		IM: ok, so I am not sharing the application anymore
			RM	FBS	IM: okay guys do you see the screen?
			RM	FBS	IM: okay M do you see those arrows a top of the slide where it says slide number 2?
ACK		B213: yes			
			SI		IM:… I like to give you guys privacy.
			FBS		IM so, any questions before I leave?
ACK		R230: yeah question	FBG		IM: okay R shoot the question
ACK		B213: ok			
HES		R230: which inspection form are we supposed to fill out?	HEG		IM: oh, the inspection form, you have one that you had to feel out before the meeting meaning that you have checked M's code before here, and now during the session you want to go back into Laulima and in the same place where you found that one, there is another one that is the group inspection form. And only one of you can fill it up, and that has to either B or R and only one for the whole team. So, let's say B, you can co back in there and open the [*] B, you actually filled up the individual form.
			OW	RM	IM: You guys are going to fill the group inspection form, the idea is that you are going to fill it up according to everybody's opinion, you know because the first time you filled it up, you actually filled it up only with your opinion but if you go into the group inspection form, I think I left it open for you. Can you see it B, in Laulima?

PRI	SEC	Chat	PRI	SEC	Talk
			ME		IM: let me just double-check that its open for you guys
RM		B213: sorry, my volume is giving me problems.			
SDR		R230: crud I think I opened it already.	IA		IM: okay, it should be okay. Either one of you can open it. R or B can just fill it up with all the consensus opinions about M code. You have to do it together, because perhaps, one of you thought the quote was really bad for example, and then after you met with each other you realized, wow! The code wasn't that bad its actually pretty good, I just wasn't looking at [*] so you go and fill it out as a consensus, something you can all agree on.
HES		B213: What about this inspection form			
			FBS		IM: Ok, so is this ok? Does this answer your questions?
ACK		R230: Roger that	SI		IM: ok, you sound like a videogame player. Ok B
SI		M212: LOL			[*]
SI		B213: LOL	NTA		… ok, then I will see you in ten.
SI		B213: o.m.g.			
ACK		M212: ok			
ACK		B213: ok			
HES	RM	B213: what is going on?			
RM		B213: I could not hear anything			
NTA		R230: I've got the form we need to fill out with feedback			
ACK		B213: oh okay.			
ADE		R230: just let me know what you want in it.			

PRI	SEC	Chat	PRI	SEC	Talk
FBS		M212: that's only for u two right?			
FBG		R230: yup			
FBG		B213: yes			
RID		B213: R just go down the list.			
SKN		B213: no magic numbers were used. Formulas worked and everything that was asked for was in the code.			
ME	FBS	B213: how are you doing R?			
FBG	RM	R230: sorry, too many screens open. I'm just going over the list and it's the same as the individual form.			
ACK		B213: ok			
NTA		R230: my comments are pretty much the same, and I've added your comments about the magic numbers and such.			
FBS		R230: anything else you want to add?			
FBG		B213: no thanks that's it.			
RCW		B213: because I know Blanca wants us to make sure we follow her directions.	FBS		IM: hey guys, any questions?
ACK		M212: no			
ACK		B213: no			

PRI	SEC	Chat	PRI	SEC	Talk
AGR		R230: nope just filling out the form now			
			RID	EMS	IM: only one of you guys should fill out the form, okay! you guys are done. And I thank you for it.
			FBS		IM: Do you guys feel like you learned something here? Talking to each other?
FBG		B213: yes	ACK		IM: Good then, great.
FBG		B213: a lot			
FBG		M212: yes			
			SKN	RCW	IM: the whole purpose of this thing is that you get to know each other and that you share and that you know that there is more than Blanca to ask you know? [*]You can ask to others it's kind of hard in an online course but now, you know each other.
SI		R230: yeah, quick question actually, can I use Elluminate anytime?			
			SKN		IM: I think so, I think you can come in. You know, [*]just make sure there's nothing at the time that you want to get into Elluminate in case any meetings are going on. And yes I don't have to be there you can use it as a chat room.

Figure D3. Session 1–Team 3 Transcript

PRI	SEC	Chat	PRI	SEC	Talk
		Nov 3, 2010 at 8:58 PM			IM: (Gives instructions on how to share the screen).
			OW		[+]Explaining how to showcase the code, highlight, explain why that part of the code was done.
FBS		D323: type as a comment in the code?	FBS		IM: So any questions about how this is gonna go?
FBG		B321: None	FBG		IM: D, you do not have to explain each line, for example in here you can highlight all the part that says final blah, blah, blah and say: These are my constants. Then, go on and highlight your variables and you move
FBS		M312: Can you explain	SKN	FBS	IM:Maybe what you have to explain a little bit more is when you do the try/catch and when you do validation and things like that, ok? Whatever they ask you, ok? and about the paper that you have to turn in, ammm I think that you should have submitted it by now, the individual inspection form and the group inspection form is exactly the same its going to be * in Laulima. I have to open it while you are logged in. So I have to go back and open it for you and once it is open you will be able to see it. Ok, Is that… does that answer your question?
FBS		M312: The two sheets to turn in?			
FBS		D323: So I got to explain each line?			
SI		B321: D is going to fall asleep if it's each line			
			FBS		M312: I seem to know in the assignment, when I was looking at it before that I can turn in the paper after mi BMI assignment has been graded and when I was looking at it yesterday, I didn't turn in the personal one because my BMI hasn't been graded.

PRI	SEC	Chat	PRI	SEC	Talk
SI		IM: Or you will fall asleep	FBG	RCW	IM: Your assignment was graded like a week ago. I left the grades up for like 5 days except for D's. I emailed everybody and say that the code that you guys were grading, the code that you were looking at, I was retracting so that you could give your own feedback not my feedback, because it is not fun that I put all the feedback and then everybody goes oh she said that so I'm gonna say the same thing. And then, two days ago, three days ago, before the sessions began, I retracted all the grades from everybody, so that you cannot go an say, oh "blanca said that about my code so I'm going to fix that" you know, I want you guys to give each other feedback not to use my feedback that's why.
SI		B321: Guilty as charged			
ACK		M312: I see			
RM		B321: I can't hear anymore			
FBG		M312: no one's talking	OW	RM	IM: … I'm sorry I was talking no one was… no! I was disconnected. What I'm going do, is that I am going to mimic, what you guys, what we would do in the classroom, ok? So I am going to leave you alone for 10 minutes and then come back in 10 min to see if you have any questions and then I'm gonna leave again and come back in 10 minutes and so on. That way it's gonna be like I'm going around the classroom looking at other teams and then I come back and see if you have questions. Ah, now, before I leave, do you have any questions?
FBS		D323: So, I don't review my code?	FBG		IM: Ah, no D, you are going to give them. You are going to tell them what you did and why you did it and they are going to tell you, they are going to comment about your code. You are not

PRI	SEC	Chat	PRI	SEC	Talk
					going to comment about your own code, they will.
ACK		D323: ok	FBS		IM: Any other questions guys?
ME		D323: so who is first?	FBG	IA	IM: D, you go , you explain, you highlight, you explain and ah, they can butt in with their questions any time they, they have a question, ok? And, or maybe they let you talk or type, right and then at the end they ask you questions, ok?
			FBS		IM: Is that ok for B and M?
TLR		D323: ok. So I realized I made some mistakes	ACK		B321: Yeah, that's fine
			ACK		M312: Yeah, I guess so.
			SI		M312: Ammm sorry about that.
			SI		M312: A car is outside distracting me
			SI		IM: Ok guys, I even heard the car, it is distracting me too (laughing).
			NTA	IA	IM: Ok I'm gonna leave you alone then and I will open the Laulima assignment only for B and M but not for D, ok? He doesn't have to do it, only you guys. Ok, I'll leave you alone then, and I will see you in a little bit.
FBS		D323: so I don't do a group one?	FBS		M312: WAIT! Wait wait wait!!!! So, me and B will have the [*] to turn in?
			IA	FBG	IM: Yes sir. I am going to open it in Laulima. It is exactly the same as for the individual form, so, just agree on the answers and do it. So maybe, for example, you wanna be doing it while you are asking questions, ok? [*]
			OW	FBG	IM: because he is in UNIX. You can ask him to compile, you can ask him to change something, you know, you can ask D to do things with his code, you see? Change this, add this down and things like that, and he will do it, right? Or say, oh that is so wrong, I cannot do that, you know?

PRI	SEC	Chat	PRI	SEC	Talk
ACK		D323: I will try	ADE	ME	M312: Ah, so basically we can just go down the questions on the paper and ask D questions about it that will answer the paper.
			FBG		IM: Yes sir, indeed. And, or you can see because you… he is not gonna tell you. For example " the code is complete?". And you ask D Is the code complete? Of course he is gonna tell you that it is complete and its perfect and super cool, right?
TLR		D323: I know I have mistakes.	FBS		M312: I see, so, its common to see same answers in the paper, like exactly the same?
EMS		B321: You should see mine L	FBS		IM: You mean the same answers, you mean you and your teammates or you before and after?
			FBG		M312: Me and my teammates, the group.
					[talking at the same time]
			OW	RCW	IM: The group paper you are going to submit only once so B does it and that's it. With one I grade the three of you.
			ACK		M312: Ah!, Okay, thank you.
SI		B321: rock paper scissors?	OW	NTA	IM: And now I'm gonna go. If you have any questions guys. You se how he has the application sharing? You can go back in to see the rubric that I gave you and you can see the presentation that I showed you. In case you need to see, ah! What do I do here?
TLR		M312: I didn't do mine, I misunderstand			
TLR		M312: misunderstood			
SI		D323: nice exhaust system			
FBS		IM: all ok?			
FBG		D323: all ok			
SI		M312: annoying kid			

PRI	SEC	Chat	PRI	SEC	Talk
		outside			
SI		IM: J			
SI		B321: yes			
SI		M312: revving their car	FBG		M312: all right
NTA		IM: oky me leaving	FBS		M312: Ah! And the recording, is it going to record everything? Even our talking and chatting
ACK		B321: Thank you Blanca	FBG		D323: I'll say yes
			ME		M312: All right! Lets keep it "PG". Let's go!
ACK		D323: I presume so			
ACK		D323: Im recording to			
IA		D323: ok firstly	HES		M312: How do you record again?
HEG		D323: oh no it gave me an option as a moderator			
			ME		M312: MSN?
			ME		M312: Sorry I do not have MSN, I have Skype
			IA		B321: [*] Do you want me to pull the form? Or do you want to? I don't care.
			TLR	ADE	M312: you have experience on it, do you want to do it for tonight only? Because I never got to do mine because I misunderstood, I thought we were turning it in today because when I looked at it yesterday I had my assignment graded, so I was waiting for it to be graded.
			NTA		M312: Ok, I will go ahead and fill it out then, ok thanks, eh? I'm gonna go ahead and pull it out too but so I can see the questions and I know what to ask.
			ME		B321: Ok she didn't open it yet so [*] for now we can just view the code

PRI	SEC	Chat	PRI	SEC	Talk
IA		D323: Ok, I'll start with the mistakes I made	FBS		M312: It is the paper that is called "group inspection"?
FBG		D323: its in PDF format	FBG		B; yeah, under "tests and quizzes"
			FBG		B321: Ok, not the one under assignments.
FBG		D323: it's the same form	RM	FBS	M312: am I disconnected?
FBG		D323: I can hear you	ACK		M312: ok thanks!
			GS		B321: Ok, [*] whenever you guys are ready.
IA		D323: ok, ill start	NTA		10; ok I will look at the form too, I will look at the form from assignments though and so I know what questions to ask
			FBS		M312: so, its exactly the same question, right?
			FBG		B321: yeah
			ACK		M312: all right.
			RID		M312: does the code compile? Can you compile the code D?
ACK		D323: Sure	RID		M312: No! I mean, you show us, like, compile it now? [jiggling]
ACK		D323: there you go			
			FBG		M312: ah, ok, you are doing it, so it compiles
			RID		M312: can you go ahead and run it?
			FBG	GS	B321: looks good, output is correct. On to the second question.
		After running the code, they decide that the output is correct. (must snapshot at 14:31).			
NTA		D323: ill do wrong input too	ACK		M312: [*] java BMI [laughing]
EXP		D323: This is where I made a mistake	SDR	FBS	M312: Oh! Wow! It doesn't break?
			FBS		M312: Don't we want it to stop if they have an invalid one in there?
FBG		D323: Yeah			
SDR		D323: My logic	COM		M312: Oh ok. I did mine correctly,

PRI	SEC	Chat	PRI	SEC	Talk
		was wrong			but…. [*]
FBS		D323: should I go back to the code?	FBS		M312: hey B what number are we on right now?
			FBG		B321: number 2
			ACK		M312:ok
			FBG		M312: So, I guess we can put single error found or multiple error found? Just because of the try logic
FBG		D323: so im not the correct way to do the try and catch statement	FBG		B321: Yeah I put single error found, incorrect try logic.
			ACK		M312: ok
FBG		D323: also have an error with one magic number			
FBG		D323: with dBMI = 25			
			RID		M312: can you scroll down and show us that number dBMI…. Oh! Okay
			NTA		B321: I added that to the number 2 as well
TLR		D323: I only realized afterwards[1]	ME	FBS	M312: oh! Wait, wait wait! Am I writing here or are you writing it?
			FBG		B321: no! I am doing it
			SI		M312: ah, ok. [joke mumbling] just kidding…
			RID	TLR	M312: can you scroll up slightly, I just want to see what is going on… slowly.. ah.. oh! I should have opened up my code.
FBS		D323: any suggestions?	FBG		B321: I can't think of any at the moment
			ME		M312: wait a minute!, If you look at the try right there
			HYP		M312: I'm looking that there should be an else in there, after the if [*], yeah that try. The nested if in there. Where is my code? Let me look at my code.,

PRI	SEC	Chat	PRI	SEC	Talk
			TLR		M312: yeah that if shouldn't there be an else in there saying that, ah, saying that if it is avalid entry. If it is a number between the 48 and 80.
			TLR		M312: but, wait, no, no! never mind
			TLR		M312: should there be an else there saying that if
			RID		(giving directions to D into the location for the requested else)
			HYP	FBS	M312: Should there be an else down there saying that if it doesn't go, I mean, if it doesn't fall between 48 and 80 inches that is wrong or something in that form. Because the try/catch will only work if it is not a number, right?
ACK		D323: That is true	FBG		M312: so, then if you put a number in there, and it doesn't fall in between 48 and 80, it will be else then "type in the correct number" or something like that.
AGR		D323: Sorry trying to remember to do an else statement	RID		M312: else, you will need the open parenthesis after the else, and then, I mean, no, not parenthesis, that bracket thing, the open bracket and then I guess you can go down one line and then do a System.out.println of "please type in a number between 48 and 80 inches"…. Yes , or something like that. I don't have my code open up right now, but yeah.
FBS		IM: Any questions? Or can I go away and come back in 10 more min?	FBS		M312: B, what do you think about that?
FBG		B321: good for now	HYP		M312: cause I figure that the try only catches if the input is not a number.
NTA		IM: ok, leaving.	ADE		B321: Yeah, that's true. And he has to capitalize the "System.out".
			TLR		M312: yeah the capital "s". I don't know if that mattered. I haven't tried it that way.
			ACK		B321: yeah, me too.

PRI	SEC	Chat	PRI	SEC	Talk
			RID		M312: then close the else. Put a close bracket after the System.out.println… yeah, after that
			RID		M312: yeah I mean, before that… Maybe that is the same problem in the… can you scroll down in the other try? Or you can try compile
			ACK		B321: yeah
FBS		D323: Try compile?	HYP		M312: see if it works. I believe that its gonna work though.
			HYP	TLR	M312: maybe there is the same problem down in the other try. Yeah!
			RID		M312: Can you scroll down to the other try
			TLR	RID	M312: expected! The close bracket is in the wrong…this if is good… the else is good. There is an extra bracket. I guess if you could remove that bracket from there. The one that is close to the else.
			FBG		M312: I think that's great
			IA	RID	M312: can you try that?
ACK		D323: ok	FBG		B321: This is great man! You should have done [*]
			SI		M312: oh God!
			FBS		B321: what is going on in here? [laugh]
			FBG		M312: we are ruining it
ACK		D323: crap	IA	RID	M312: can you pico into it again?
			EMS		M312: Sorry man! I hope you have a backup of this program
			SDR		M312: [*] ok so it was the closing for the try..
			IA	RID	M312: can you put a closing bracket underneath the closest try?
			SDR	TLR	M312: oh wait, wait, wait, no nevermind that's gonna close the try, nothing is closing the else.
			TLR	HYP	M312: no, I mean that's gonna work too, I guess you can put closest try and then yeah… and that's gonna close the else. I think that is what we did earlier.

PRI	SEC	Chat	PRI	SEC	Talk
			IA	RID	M312: Can you try that out?
			TLR		M312: (with lower voice) that closes that… that closes that [*]
			TLR		M312: oh no! ok, ok!
			FBS		B321: Is that semi-colon where it says between 48 and 80 inches
			ACK		M312: yeah, the semi-colon
ACK		D323: ok I see	NTA		M312: Oh, ok, there you go
			TLR		B321: ok, that should be good.
			SI		(mumbling about "last night")
			FBG		M312: oh! That's great, that's great!
			FBS		M312: oh! Should we make it so that it closes in there. It closes if the answer is wrong?
FBS		D323: shouldn't it shut down the program?	FBG		B321: Yeah cuz if not [*]
			FBS		B321:you put your height right?
			FBS		M312: I forget what to do for closing code. Is that break?
			FBG		B321: I'm not sure
			TLR		M312: I'm not home right now so let me see if I can find my code. That would be great if we could just close it right away. You know? If the input is wrong, close it right away. I guess we can put that inside the catch and inside the else. That would be great. I think that's howI did it on mine…. I'm looking at mine right now.
			FBS	RCW	B321: What chapter is that catch and try statements? [*]
			FBG		M312: I can't remember
FBG		D323: I think 13	FBG	RCW	M312: I haven't been watching the sessions. I only look at the powerpoints
			NTA		B321: trying to find it right now
			FBS		M312: ok. what are you trying to find there? The break?
			FBG		B321: yeah
			TLR		M312: I think I did it on mine like that but let me check [*]
			SDR	COM	M312: Oh! I did it differently!!

PRI	SEC	Chat	PRI	SEC	Talk
FBS		D323: nested it different?	COM	EXP	M312: Oh! Nevermind! I did mine different. I nested an if/else inside an if/else and that if/else is nested inside a try/catch and the try/catch is nested inside an if/else and the if/else is inside a try/catch. But it worked though. Every time there was a wrong answer, it would stop right away.
FBG		D323: man! That sounds confusing	TLR		M312: Yeah, I know. If I look at my code now and I do not fully focus on it, I don't think I'll understand from looking at it right away.
ACK		B321: yeah! Haha	HYP		Maybe that's why Blanca didn't choose it (laughs)
			TLR		Its so confusing when I look at it now, but it worked, and it was small.
SI		D323: My brain is running on fumes right now.	EXP		M312: I don't know if we can put break… [*] I think that's why I made it this way.
FBG		B321: I don't see anything on break			
FBG		B321: In session 13 ppt			
TLR		D323: I don't remember doing a break	RCW		M312: No, I don't think we saw it in the presentation B. I don't remember it. That is why I did it this way to completely stop it if anything, if the answer goes wrong. I think this is good for now.. ah.. yeah.
ACK		B321: oh			
			HYP	RID	M312: I think we can move on to the next try and put an else there if there's no else there. I think that would be great, yeah? Do you wanna do that B?
			ACK		B321: yeah [*]
ACK		D323: Ok	AGR		M312: so, after that if, the same like that, right there
			RID		M312: you can copy it, we can then just change the string in the "please type a number blah blah blah…

PRI	SEC	Chat	PRI	SEC	Talk
			TLR	FBS	B321: hope it doesn't crash either, it just tells you the input, the weight, if you put an invalid height it will just tell you its invalid and it will tell you to put a new weight, right?
			ACK		M312: oh
			FBG		B321: it never crashes
			FBS	TLR	M312: oh for this one? Would that count as crashing? On no wait! It wouldn't crash because of the try/catch.
			ACK		B321: yeah…
ACK		D323: yeah	AGR		M312: yes, technically it didn't crash because of the try/catch
			ME		B321: Question number three
			ACK		M312: ok, ok
			FBS		M312: I was just thinking, are we supposed to modify this code? I don't think I see it in the questionnaire.
FBG		D323: I guess its part of it	TLR		B321: I 'm not sure…think right after, then we can
			TLR		B321: you can resubmit it after
			RCW		M312: At the moment we are just helping D, yeah
			OW		B321: Actually he's just supposed to explain what he did section by section and we may have comments or questions to ask him. At the same time we try to fill out the Laulima evaluation form
FBS	RCW	D323: are we all getting the same grade for BMI?	FBS	ME	M312: ok, and that's the one that you are filling out right now, right?
			ME	FBS	M312: do I have to do it too or do I just give out some answers and you fill it out for us.
			ME	FBG	B321: yeah, just read along [*]. I'm on 4 right now. So…
			ACK		M312: ok [*]
			SI		M312: oohh! I don't wanna be mean to D! Just kidding
			ME	RID	M312: can you scroll down … can you

PRI	SEC	Chat	PRI	SEC	Talk
					scroll up a little more? [*]
			RCW		M312: I am trying to look for a good section. Something elegant and robust. So what I am trying to say is that we changed some parts [*]
FBG		D323: its all good			
ACK		B321: lol			
SI		D323: im just joking			
			RCW	COM	M312: I'm gonna be getting a grade lower than yours because I didn't print on the personal one, I misunderstood, when I looked last night, my grade, my assignment didn't have a grade [*].
EMS		D323: Im sure you doing fine in the class, you sound on top of it.	FBG	ME	M312: so the bottom part should be good. Let me look at it once. [*]
ACK		B; Im' actually not	TLR	ME	M312: I think there's something off with the…m… can I wait! Wait…
IA	FBS	D323: want to see the integers?			
SI		B321: chem+ calc+ phys+ ics = /rape			
SI		D323: B, this is my last semester			
SI		M312: you're lucky, I have to work so I can take classes			
SI		D323: I work too	IA		M312: I wanna see the BMI equation
SI		B321: same here L	RCW	FBS	B321: she said the actual equation didn't matter, right?…
			TLR		M312: Oh, I mean, cuz, Is that I was looking for the robust part of the code so I was thinking of taking out the bottom part of it.

PRI	SEC	Chat	PRI	SEC	Talk
FBS		IM: Any questions?	ACK		B321: all right
			FBS		M312: Hey Blanca, would the equation matter? Well, I don't know the equation right now, but would it matter?
EXP		D323: My equation is correct	FBS		IM: Remember that you emailed me one and then I corrected it. Are you talking about it?
EXP		D323: I derived it			
HYP		B321: If D derived it, then it should be correct.	FBG		IM: …what really matters are the error messages, validations, setting boundaries for your values…
			HYP	FBS	M312: So I guess the robust part is the bottom one, hah?
			FBG		IM: The one that matters is how you read from the user and how you validate it [+] Like if you try and you cannot crash the code, when you put different values and [+]
SI		D323: years in engineering classes come in handy	FBS		M312: About the word crashing. If you put the wrong value or the incorrect value and it still goes on, ah.. there is a try/catch so the code wouldn't crash, Is that crashing or is not crashing?
			FBG		IM: That is not crashing but its just incorrect handling. But at least it didn't crash. It is worse [+]
			IA	FBS	M312: Ah ok then, for the question number 4 we can put the bottom part "elegant and robust" B?
			SKN		B321: Ah there is actually no part where you can type, you just select a number
NTA		IM: See you in 10	ACK		M312: ah yeah, yeah… sorry about that.
			SI		M312: I don't see a "very poor", nah! Just kidding
			SI		B321: D, what is your major?
SI		D323: civil	SI		B321: oh, my plan too. At least [*] my math though.

PRI	SEC	Chat	PRI	SEC	Talk
SI		D323:cool	SI		M312: congratulations to you guys
			ME		M312: so we are on number 5
SI		D323: I graduate this semester	AGR		B321: yes on number 5 right now
			COM		M312: "Code was easy to read". I strongly agree on that one. My code was ridiculous to read. Whereas this code is pretty easy to read
SI		D323: I can't wait to make mediocre parking lots!	TLR		M312: [*] you can tell, there is multiple parts
SI		B321: just messing.	ACK		B321: yeah
ACK		D323: makes sense	SKN	FBS	M312: that's how I'd been coding anyway! Just so I can [*] for me like the try/catch will be in the same indentation and then the if else will be the same indentation but more in, you know what I mean?
			AGR		B321: yeah, ok.
			FBG		M312: code was properly indented so it was easy to match up braces. So… I guess we can put a 4 on that.
			RID		M312: "Variable names were descriptive and relevant". Can you scroll up?
			FBG		M312: [+] oh, I can tell, it looks good
EXP		D323: I should do indents, I just don't do it cause um doing it on a 14" monitor	FBG		M312: (reading) "code was properly indented so that it was easy to match up braces". So,I guess we can put a 4 on that.
			SI		M312: oh yeah! Oh wow! 14 inches! Its one of those small computers, right?
SI		D323: yeah	SI		M312: those are pretty neat specially if they have [*] video cards in there
			FBG		M312: the comments [*] were clear and helpful. Yeah, they were clear and helpful. I think it can be a five on that.
			AGR		B321: Yeah, I give him a five

PRI	SEC	Chat	PRI	SEC	Talk
			ME		M312: no, sorry I'm falling behind man!
			FBS	ME	M312: where are we now? The user interface?
			FBG		B321: "The purpose, of the program was clear"
			FBG		M312: yeah, I guess it was pretty clear, because it was asking for input and then after that input, he's trying to see it using the try and catch and then inside the try and catch he's got the if/else ahm…[*] So I think it was pretty clear
			ACK		B321: yeah
			ME		M312: and then, "there was easy to enter input correctly"
			TLR		M312: yeah I think it was easy to enter input correctly
			CHQ		B321: [*] two lines though
			FBG		M312: yeah, error messages were very helpful and explained what was wrong
			TLR		M312: I don't think it explained what was wrong, though
			TLR		B321: [*] a value in between..
			HYP		M312: after the else I guess. After we put in the else and what not I guess it made it more clear
NTA		D323: I got error	AGR		B321: yeah, its ok now
			AGR		M312: yeah, its ok [*][+]
			ME		M312: "the program was fairly easy to use"
			FBG		M312: I mean its just asking for inputs and then it just gives you the result, it was pretty easy to use
			ACK		B321: yeah
			TLR		M312: then comments… comments, coments
			ME	FBS	B321: wait a minute what question are you on
			FBG		M312: I was just looking at the work "comments" below number 11. [*] easy to use. So are we on features? 12?

PRI	SEC	Chat	PRI	SEC	Talk
			ME		M312: the extra features, if any were a valuable addition to the program
EXP		D323: no extra features	ACK		B321: yeah
			FBS		M312: can you see any extra features in there B?
			RID	ME	B321: can you just scroll through it fully?
			CHQ		M312: D says no extra features but if we can find some extra features that would be great.
			HES		B321: who can give me an example of an extra feature?
			HEG	FBS	M312: Oh yeah!, I guess something that you don't really need but that makes the program a lot easier. Maybe that's what it is, ah?
			AGR	FBS	B321: yeah… oh but what could that be? For example?
EXP		D323: I was rushed for time just did what the assignment asked			
HYP	FBG	D323: Maybe using GUI???	RCW	FBS	M312: I think that we are going to do that later on, with the applets and everything. Yeah? I guess we are good. Anything else in there?
SI		D323: how did your guys applets go btw	SI		M312: it was good, it showed and everything but in only blinked twice for some reason.
EKE		D323: general = I used magic numbers	AGR		M312: Yeah I used magic numbers too
SI		D323: I can send you my applet			
SI		B321: (sends applet link)			
SI		B321: no laughing	SI		M312: I don't see it, yeah… in my loop, my light was blinking but it was only the first seconds or something, I think that it just calculated it so fast that it didn't have time to show it. I

PRI	SEC	Chat	PRI	SEC	Talk
					even tried to do it multiple loops of infinite loops [*] and it didn't work
			SI		M312: oh wow!
SI		D323: that's quite cool	SI		M312: let me look at mine. I put mine in here
SI		D323: (sends applet link)	SI		M312: I guess I just put it at my index [*]
SI		B321: d > mine	SI		(M312 explains that his applet is supposed to blink and how he keeps on pressing the refresh button in the browser and it barely blinks)
SI		M312: (sends applet link)			
SI		BT: mine looks like some 5 min paint face eating a turd.			
SI		D323: wow			
SI		B321: /shame			
SI		B321: L			
SI		B321: yours is nice			
SI		D323: wow 10	SI		M312: no man, I'm cheating, that's how
SI		B321: Thank you 10, hopefully she grades mine before yours	SI		M312: (laughing) no man, she doesn't grade it that way as long as it works [*], I think that's what she said before.
SI		B321: the sunset is beautiful…	ADE	ME	M312: Are we done with this?
SI		B321: /tears	SI		M312: that's what you think of when you have girl problems, that's how it came up.
SI		D323: I think so			
SI		D323: B is cracking me up			
SI		B321: /blush			
			ME	FBS	M312: that comments, is that comments for the whole code. Is that

PRI	SEC	Chat	PRI	SEC	Talk
					what its asking?
			FBG		B321: comments in general about anything
			ACK		M312: ah ok.
			FBG		M312: I guess you can mention the magic numbers too, the number 25. I guess you can also put how it doesn't close if you put in the wrong input
EKE		D323: I used magic numbers	FBG		M312: yeah it was, it was a nice code. It was easy to read. That's what I'd been thinking about it
FBG		D323: it was awesome code!	COM		M312: if you read my code, its super complicated
ACK		D323: lol	FBS		B321: aside from the [*] magic numbers the program [*] incorrect answers given? Is there anything else 10 that you think needed to be changed?
ACK		D323: thanks	IA	RID	M312: Can you scroll up a bit D? … oh, oh… The io. Is that the correct io? Ah, you used the buffered reader… ok… … can you scroll down a bit more and a bit more. Oh the indented! You don't' have to do that, he is using the 13 inch laptop…. Buffers… right on D. Can you scroll a little bit more?
SI		B321: I just had a chem. 162 exam today on buffers L	FBS		M312: Ok yeah except the indentation to make it easier to read… how you did the last part, the if/else [*]
SI		D323: I remember those days	FBS		B321: shall I go ahead and submit this though?
SI		B321: /pain	FBG		M312: yeah I think that should be good.
			ACK		B321: ok
ACK		D323: ok			
FBS		D323: can I log out of unix?	ACK		B; yeah.
			TLR		M312: I was just wondering.
SI		B321: so D, any suggestions if I plan on	SI		M312: Do you guys like wanna see my code? Because it closes if I put the wrong input.

PRI	SEC	Chat	PRI	SEC	Talk
		continuing with CE?			
			SI		M312: if not is all right. Doesn't really matter
SI		D323: sure can you send it to us?			
SI		B321: can D appoint you to a mod?			
SI		D323: oh maybe I can	COM		M312: oh, I think I'm in the mood now. So how would I open… tools, application sharing…[+] ok pico bmi.java so from the top
FBS		IM: Any questions?	EXP		M312: + import java.util.*; is because I used the new scanner, I mean, I used the scanner instead of the BufferedReader [+]
SI		D323: got coffee?	EXP		M312: if you look at the code it looks super complicated but then, I wanted to do it this way so if I put any wrong input it just closes right away.
SI		IM: >-)			
FBG		D323: wow, a lot of nesting			
SI		IM: I meant J	SI		M312: you guys will get higher grades than me, I didn't turn the [code inspection] personal one.
IA	RID	D323: Scroll down			[+]
			EXP		M312: as you can see, there's no comments, so I'm gonna be off of that. I was just rushing on that [+]
			EXP		And then the constants in the naming, I didn't do dHeight, dWeight
					[+]
			EXP		M312: It's a scanner instead of a buffered reader
NTA		D323: B dissapeared	EXP		M312: And my numbering may be wrong, just because it's the word between, I always exclude the numbers.

PRI	SEC	Chat	PRI	SEC	Talk
TLR		D323: I am confused about the brackets	FBS		M312: your confusion about the brackets in general or… or about the code that I showed. I can open it again and explain it to you. Do you want to do that?
FBG		D323: its alright	EMS	FBS	M312: just let me know if you need any help, yeah?
EMS		D323: I appreciate it			
NTA		D323: wb B	FBS		IM: so how is the review going?
SI		B321: sorry, road runner owned me	FBG		M312: It was B that did it, I mean it was B that was writing it down…
SI		D323: I think so			
SI		D323: thanks you guys and lady of the house			
SI		IM: I hope you had fun			
SI		M312: it was nice			
SI		B321: D I need advice!			
SI		IM: I hope it was useful too			
SI		D323: sure what on?			
SI		B321: civil			
SI		M312: engineering			
SI		D323: email, Skype me			
SI		[exchanging Skype details]			
SI		B321: Who is XX?			
SI		D323: X?			
SI		B321: X.X			
SI		B. Added me			
SI		B saying			
SI		B321: in my ics			

PRI	SEC	Chat	PRI	SEC	Talk
		class			
SI		D323: a guy in our class			
SI		B321: Xxxx Xxxx.			
SI		B321: student ICS111			
SI		B321: oh I see			
SI		D323: ok thanks for the session			
SI		B321: thanks everyone			
SI		B321: see you next time			
SI		B321: best hour of my day			
SI		D323: technology is getting quite fancy			
SI		IM: thanks			
SI		D323: bye!			

Figure D4. Session 1–Team 4 Transcript

PRI	SEC	Chat	PRI	SEC	Talk
SI		IM: hi			
SI		K411: Hi Blanca			
SI		IM: sorry, I was caught at the polls			
SI		A432: hi			
SI		P423: so that's what it was			
RM		P423: yup I can hear	RM		IM: did you guys hear me? I just want to make sure that everything is okay. I can see that you have already been working it out.?
RM		K411: I can hear you fine	SI		IM: I apologize, I came here running,
			FBG		IM: do you guys have any questions that I can help you with
			OW		IM: let me allow more than me to talk.
					(The instructor explains the rationale behind allowing more than one speaker, and explains the process)
			OW		IM: Actually what you were doing is the right way to go. A will explain her code, you know. After she is done explaining to you what you did then what you guys can do is to give comments and say what you did different and what she could have done better and what she did better than you, you know?
TLR	RCW	P423: I was saying earlier that I wish I could see. My evaluation thingy that I did. Because I listed out everything that I was going to suggest during this			

PRI	SEC	Chat	PRI	SEC	Talk
		meeting.			
			HEG		IM: so you want to see your evaluation thingy. I think I can pull that from Laulima. Or let me see if I can go back in Laulima and allow you to see it again. Just hold on one second I am going into Laulima.
TLR		P423: yeah, good to date hasn't passed yet, so doesn't let me review it.	IA		IM: In the meantime while I get P's evaluation, if you guys can continue your discussion go ahead I don't want to interrupt you. Let me go into Laulima and check for the code…. For the evaluation I'm sorry
ACK		P423: thanks!	ME	IA	A432: since Blanca just started recording I will go back and start from the beginning. That way it will in the recording, and we can go back and review it if we need to.
			TLR	COM	A432: [*] it's just that in my code, in the height section, when it ask the user for the height, it would, if it was an incorrect height, it wouldn't continue to ask for the correct entry for the height [*] it will continue on asking for the weight, and then, after that in the BMI calculations. It was just taking, you know, anything that was given.
			OW		IM: What I am going to do right now is that I am going to allow you guys to see the Laulima.. so what I am going to do with P, is that I am going to make you a moderator as you become a moderator you can do a few more things and actually I'm gonna have to make A a moderator as well because she is the one that is showing her code

PRI	SEC	Chat	PRI	SEC	Talk
					and what I am going to do P is that I am going to give you control of my desktop so you can look at Laulima…[+]
ME	IA	P423: okay, Iine 2, there was that A brought up.			
			ME		IM: so feel free to move it around
SKN		P423: So what I did was a little research on the net and found out about the system.exit command.			
RCW		P423: so probably not the way we were intended to resolve the problem, but it worked > _ >	COM		K411: I mentioned the system.exit is a lot nicer to code than what I did
			TLR	SKN	K411: like I said, the massive nesting of if statements, it eventually got to the point that I went to trial and error where I could get it to close in case that something invalid was entered, or keep going if something correct was entered.
TLR		P423: could either do it that way, or use some sort of Boolean loop.			
TLR		P423: where it tries again until there is a valid input.			
FBS	ME	IM: do you want to keep this Laulima page up?			
			TLR	COM	K411: one thing that I must say is that A code was much, much more organized than my coded and it was easier to read. So, kudos for that.

PRI	SEC	Chat	PRI	SEC	Talk
RM	FBS	P423: Uhh, is there any way I could copy that stuff on your screen and somehow pasted into mine? XD			
FBG	EMS	IM: just try. I'm sure you can			
ACK		A432: thank you!			
			EXP	HYP	A432: this is the part where if the user wasn't entering the correct height, and then it will continue on, [*] I guess that someone mentioned a bunch of if and else in there, is that correct?
NTA		IM: I mail it to you	SKN		K411: yeah, I basically had to set up where it was a series of if-else statements where if they did it correctly it would move on, otherwise it would just skip all the way to the end and say that it wasn't correct, and ended of the program.
EMS		P423: sorry for all the hassle XD/noob			
GS		IM: okay you guys go on. Sorry.			
FBG		P423: anyway, I would suggest using either the system.exit or a loop to fix that problem			
FBG		P423: other thing I remember seeing… The 703 in the BMI formula is used as a magic number	AGR		A432: I'm not familiar with the system.exit program, but I guess it sounds like, that was also [*]
			FBS		A432: I guess, Blanca, is there a way that all the members, you know, if I can [*]we can pick their strengths from their code and then the strengths from my code, and together put out a better program.?

PRI	SEC	Chat	PRI	SEC	Talk
RID		P423: 703 should be made into a constant.			
			FBS		IM: A you asking if you can get their ideas and incorporate them into your program somehow?
			RCW	TLR	A432: yes, [*]for me I think I would have learned a lot more if I get to look at the 2 different ways that they did it and then just learned what I have learned so far it, and then I can kind-of combine it, will give me a better perspective, you know, get me more ideas on different ways in which I can improve my code. I know that we are doing this as a team effort too, I was just working on my code and find, that I think for me I can learn a lot more if I can see the other ways it was done too.
			OW		IM: I cannot have you all sharing everybody's code, because it will take you a very long period of time. And this way, you show your code and they give you the feedback then you can go ahead and fix it. As they look at your code, and they get ideas for their own code. So basically, what I'm telling you is that the

PRI	SEC	Chat	PRI	SEC	Talk
HEG		P423: I'm willing to send you my code, but yeah, it might get a little swamped for this session if I try to put it up here. >_>			purpose of this, [*]what you're asking for it to happen, it's what's going on right now. Like sharing the code and then sharing ideas the only difference is that you're not looking at their codes directly. What you can do right now, and what the other teams did is that you can share your desktop. So the same way that I was sharing my Laulima. What I would like you to do A is to bring up your code in UNIX. Just bring it up in your PC, and then I'll tell you how to share it, that way if K or P have an idea for you, you can go ahead and fix it and you can compile it again and see what happens. You can do it in real time and you can be getting the feedback and fixing it. Now, what the other guys can do, they have their own code, where they can type in some comments of the idea that they are getting from the team meeting and then they can fix their code. So after this is done you can resubmit your code with all the improvements and all the little things or big things that you have learned from the team. I don't know if that answers your question.
			FBS		A432: yeah, I got that, but I guess that if maybe [*]that if-else statements because that's what I have trouble on. And if you could just expand on that so that I can get a better understanding on how you did yours
			SKN		K411: yeah no problem, instead of doing multiple try and catch statements. Basically, just one giant try and catch, where the

PRI	SEC	Chat	PRI	SEC	Talk
					whole thing is inside one thing so it is a try, and then I have them enter their height. And after the first line "please enter your height" [*] read it, came if they enter the height that is in the parameters, then it says "please enter your weight" and then it reads the weight and there is another if statement that if they enter the weight in the right parameters then it gives them their BMI and there is another if statement there that it says if the BMI is high, over 26 then it will say it is considered high. Then all the else statements try to catch up. So after the last if statement there's a high BMI the else is Otherwise it just says that your BMI is good. So if the BMI is less than 26 it will say that the BMI is good and the else statement for the [*] is about weight, comes in and says that if they put in a weight that is valid a weight that is not valid it will just say "sorry that is not a valid weight" and that ends the program there and after that is the else for the height and says the same thing, if they do not put the height that is valid then it will say "sorry that's not a valid height" and then the catch for the try comes in after all the elses and if they enter anything that is not a number it will say " that is not a number" and that way you do not have to do multiple try and catches, ammm…

PRI	SEC	Chat	PRI	SEC	Talk
SKN		P423: I had two verification each for the height and the weight.	HEG		K411: if you nest all the ifs inside that one try with one catch at the end then anytime that it is not a number it will terminate the program right there. You do not have to worry about trying to constantly put something there in the program.
SKN		P423: The first verification was a try/catch to see if it was an integer			
			AGR	RM	IM: Ok I think I see what you did and I was going to ask K to write it but I have a problem here with the screen. Let me load something so you [*][+]
			FBS		IM: Is that ok with everybody?
ACK		P423: yup			
ACK		K411: no problem	GS	NTA	IM: Ok, share on then! I will see you in a few minutes
ACK		P423: thanks	SKN	AGR	K411: So, what Blanca wrote there is pretty close. Basically it is all nested, one try statements and then you tell them to enter whatever the height first. [*] the height first and then you begin with the if statements and each if statement says ammm if they enter something in the correct range then it moves on and the corresponding else statement says otherwise you give them some kind of error message and end of program there. And you do this for each one, so you have one in for the height and then you have a [*] if it is correct and then you have one for the weight then you give them the BMI if it is correct then you have an if for if the BMI is high you tell them

PRI	SEC	Chat	PRI	SEC	Talk
					it is high and each one has a corresponding else statement saying an error otherwise with the exception of the BMI which has an else statement saying that the BMI is good cuz [*] is good…
AGR		P423: yeah, I get an idea of what you did	AGR		A432: Then you put all of it in one try/catch statement [*] (laughing)
			SKN	TLR	K411: yeah, I'm really lazy, so I look for the easiest way so I can have as few things as possible and that way it is sort of one giant net for when anything that is not a number, it just kills the program there and gives them an error message.
			COM	EMS	A432: ok thank you, that helps a lot because I am trying [*] spreading it out but it may be too much, nesting it in one sounds a lot better. I mean, it is like a straight line
			SKN		K411: yeah I think, I acutally spent… the hardest part was trying to format it all the indentation correct I think it was about half an hour just trying to line everything up and making sure that all the [*] was correct and all the indents were correct. It was kind of a mess.
IA	HEG	P423: Also, another thing I would suggest adding as a courtesy to the user			
HEG		P423: if they need to input in a specific range, should tell them what the range is beforehand			

PRI	SEC	Chat	PRI	SEC	Talk
NTA		P423: For instance I did this	AGR		A432: yeah, that's a good suggestion to let the user know the ranges [*] you know, with "incorrect entry" maybe there will be more aware of what they are entering.
			AGR		K411: Yeah, at the very least, like you should do it [*] in the error message tell them you know, you have to enter a height or a weight between this and this. I just said [*] is not valid because I [*], I was very tired and I did not think about other people but yeah it makes sense to at least give them some kind of idea of the range from somewhere.
SKN		P423: What that comes out as is			
SKN		P423: Please enter a whold number for height inches (48-80);			
SKN		P423: It lets the user know what they allowed to enter in, before they have a chance to put in "25" or something			
EMS		P423: \u0028 = (			
EMS		P423: in case that wasn't obvious > _ >			K411: sorry, I'm, sort of going back and forth between this and my code. Make [*] changes to my code as we go.
RM		P423: people still there? o.o	COM		
ACK		A432: yep, still here			
FBG	RCW	P423: another thing with the code, which I'm not sure it was mentioned already			

PRI	SEC	Chat	PRI	SEC	Talk
FBG		P423: was that the BMI was always reported as "good" regardless of what the BMI value was			A432: yeah, I got stuck in the BMI stuff down there, I think I have some ideas [*] I was messing with the other stuff about the height and weight [laugh]. I think the formula itself is wrong, so , if I'm not mistaken, I guess after the weight, I need to, I guess I need to [*] that whole thing. I'm not thinking and also I needed to put the range to recognize that anything above 25 is not good.
			EXP	TLR	
			AGR		K411: yeah when you are [*] the BMI formula, one thing you need to make sure of is you have to switch the weight and the height [*] because you also need to make sure that the nominator of the formula is height squared and I don't know how to do squared so I just put iHeight*iHeight.
AGR		P423: Yeah, that's what I did too			
AGR		P423: height * height			A432: I was wondering if you could do it like square with you know that carat sign that's above the 6 and then put the 2. I could be wrong but [*]
RCW		P423: IDK if height^2 would have worked	HYP		
TLR		P423: Didn't think to try that at the time >_>	RCW		K411: yeah, I would assume that's what it is, but ah.. I wasn't sure and I [*] I stay with what I knew so [*] I was doing it really late at night. I was going against the deadline. [*] That was before we got extended
			EMS		A432: I know the feeling [*] something my brain doesn't even [*] already.

PRI	SEC	Chat	PRI	SEC	Talk
			AGR		A432: yeah, I guess it is easier to do it that way instead of having to keep going back and forth compiling and then [*] and then [*]
SKN		P423: what I did, and this was probably a cumbersome way of doing it			
SKN		P423: was I set up two constant values, one that specified the point wher a BMI is "low" and one where it is "high"	FBS	ME	A432: so I guess, is there anything else that we can [*]if she's there or anything else that you guys wanna share?
			RCW		K411: I'm glad I got to look at your code because it helped me out. [*], I didn't assign everything to a variable or a constant I just had a lot of random numbers from when I had to put on the range and stuff, I would just throw in the numbers instead of the [*] so I lost points there, so good to see somebody else's formula, or code so I can see what's going on there.
SKN		P423: and then after calculating the BMI value I put in there if statements			
			AGR		A432: yeah that's where I was coming from when I was asking you to see the team member's code I just need to visually see it. For me, it is good to see things so I can find the strengths about that and then I can enhance what I have (laughing)

PRI	SEC	Chat	PRI	SEC	Talk
			HEG		K411: yeah, I sent some email that had my original code and I have made some changes but it has the original code in it so at least you can look at it and get an idea of what I was talking about with all the nesting of the if/else stuff.
			ACK	EMS	A432: Ok thank you, I really appreciate that.
			HEG		K411: no problem. I don't have any idea if the formatting is gonna hold through he email so you can let me know if it does or doesn't and then I can just attach the file and send you the actual java file. So you can look at it the same way Blanca sent us yours
			AGR		A432: I got it! That's ok. This is really helpful for me too, you know to actually see it. Looking at [*]
		P423: something like that >_>	FBS		K411: I didn't notice if there was a value that was considered too low. Was there something [*], I just didn't see it or did you do like independent research on the [*] assignment?
SKN			ACK	SI	A432: ah! No (laughing) by that time my brain was useless (laughing) I kind of [*] when I came to the BMI part
			ME	EMS	A432: Ah! Were you talking too P? I'm sorry!
			ACK		K411: No problem!

PRI	SEC	Chat	PRI	SEC	Talk
			RM		A432: I'm trying to read what P is trying to but I guess, my screen only shows two lines of code at the time so I have to keep scrolling up and down. Is your screen the same size or?
			RM		K411: I get about 5 lines of text at the time
RM		P423: Mine is a little bigger o_o about 10 lines	RM		A432: ok, I'm trying to stretch the screen out and it's not doing it
RM		P423: I'm sorry for having to type x_x I'm trying to use the whiteboard more			
NTA		P423: but yeah, I've covered my main suggestions, I think	FBS		K411: are there any more areas of the code that you want help looking at [*] everything I have
			FBG	RCW	A432: no, that was just my major issues that I had. I don't know, I guess, that's it. I don't know, are we supposed to somehow do the inspection as the group. I am not sure how we are supposed to access that. How are we supposed to get that. [*] the group inspection, I guess it's on Laulima but [*] about getting it together, I am not sure how we are supposed to go about doing that
NTA		IM: I'm here in case you have any questions			
			RCW		K411: yeah, I think that Laulima shows it as closed it opened yesterday between 12:am and 10:30 pm
			NTA	FBS	IM: Do you guys need me to open something? Let me know. I'm here to help.

PRI	SEC	Chat	PRI	SEC	Talk
			FBG	HES	A432: I think we are pretty much done but I think we are trying to figure how are we supposed to go about completing the group inspection form?
			OW		IM: oh! Hold on, the thing is that I was going to open in right before your session and guess what? I was here not on time so let me get it back… the thing with the group inspection guys, it's exactly the same one as the individual one ok? Now the only thing is that in this case, probably after you have the opportunity to talk to each other and figure out, and see what other people think you know, you probably have a better idea of the answers to maybe something made you change your mind and you decided that instead of being super bad code it was super good or it was medium or you realized something that you didn't know, so that's why it's a new one. So What I am going to do is that I am going to open it now and I can only open it for P and K because it is supposed to be open only for the reviewers not for the code author so A would not be able to see the group inspection form. I mean actually if you want I can open it so you can see it but the point is that the other ones are the ones that have to actually fill it up.
			IA		IM: so I am going to open it now. And it is exactly the same so if you guys feel that oh yeah this is what I did and that's exactly the way it should be that's fine but you two have to decide and can fill that up.

PRI	SEC	Chat	PRI	SEC	Talk
			FBS		IM: Any other questions?
		P423: Nope I'm good	ACK		A432: No, I don't have any questions.
ACK		K411: Nothing here			
ACK					
			NTA	ME	IM: Ok, I'm opening it now. When I tell you is done then you know you can go in Laulima, ok? Hold on a second.
			NTA	IA	IM: its open for P and K if you guys want you can fill it up while you are still connected that way you can[+] You wanna check P and K to make sure its open?
			NTA		K411: Yeah I got it. I have it open on my [*]
			FBS		IM: Ok, any other questions?
			FBS		A432: I guess I just I have one more question. I guess since you are recording, Is there a way I can go back to this if I needed to?
			AGR		IM: indeed!![+]
			NTA		IM: let me actually stop the recording now.

Figure D5. Session 2–Team 1 Transcript

PRI	SEC	Chat	PRI	SEC	Talk
		Setup and the instruction gives last minute procedure clarifications			
		One team member is still missing.			
		Setup procedure takes place and the instructor lets students know that she will be back in 15 minutes			
			FBS		IM: I would like to know if you have any questions
SI		M312: What's for dinner?			
SI		M212: LOL	SI		IM: I still haven't cooked dinner, I just came back from Leeward. I came running to my computer (laugh)
SI		M312: lol, same, just got home from work			
GS		M312: ok lets do this	SI		IM: maybe we can have an Elluminate session at the restaurant?
SI		M312: yeah! That's neat!			
		The instructor reminds them of the use of the Elluminate tools, they finish getting ready for the session.			
NTA		IM: I'm leaving and I will be back in 15 min	GS		M312: shall we start?
			FBS		M312: Anything else Blanca that you want to say before you go?... I guess you're gone…
			IA		M312: ok, let's start then
			FBS		M312: ok, you guys want me to do anything with my code? Oh no, no! wait! Wrong one! Oh! This one, yeah! [+]

PRI	SEC	Chat	PRI	SEC	Talk
			FBS		M312: Do you guys have any instructions?
			RM		M312: Can you guys hear me?
FBG		K411: I actually didn't see anything wrong with the code			
ACK		M212: yes			
FBG		K411: I played your slot machine like 300 times and couldn't break it	AGR		M312: yeah! I like to check it before I turn things in.
FBG		K411: But I tried my damndest			
ACK		M312: LOL			
EXP		M312: There were times when I could win real fast			
EXP		M312: Or almost never			
AGR		K411: Yeah, same with mine			
SKN		K411: Took like 70 tries to get my first win			
			OW		M312: So I think that either one of you have to answer the quiz on Laulima. Someone has to do it and the other person can help answering it.
ACK		M212: Ok			
			ME		M312: so who's gonna answer the quiz? It has to be one of you guys.
ADE		M212: um K you wanna do it?			
EMS		K411: I don't mind doing it if M doesn't want to.			
ACK		M212: lol			

PRI	SEC	Chat	PRI	SEC	Talk
			ME		M312: its pretty simple questions, its just.. you're just [*] your opinion. You are just answering from your opinion. You can ask M, you can tell M what is the question and then ah… he can help you answer it. I think it's the same thing as the individual one
ACK		M212: hay			
AGR		K411: Yeah, the questions are all the same			
			GS		M312: all right! Let's do this!
			FBS		M312: Do you guys have any questions on it?
ACK		M212: no no			
ACK		K411: Nope			
COM		M212: ur code looks nothing like mine			
SI		M212: lol	FBS		M312: Really?! Did your code worked though?
FBG		M212: no I don't even understand how to view it			
HES		M212: I need a tutor	ACK		M312: oh, ok!
SI		M212: lol	HEG	EMS	M312: I would help you out if I'm not busy, I'm kind of busy, that's why. So… but I would help you out if I can.
ACK		M212: yah!	HEG		M312: But I mean, you got my code, so I think if you look through it, I think its self-explanatory. There are some comments in there. You could compare it a little. How far did you get up to?
SKN		M212: I did the whole thing blind though			
FBS		M212: If you know what I mean	FBG		M312: I don't know what that means (laughter)

PRI	SEC	Chat	PRI	SEC	Talk
			RCW	FBG	M312: I mean there were some examples that you could look through, in the material. In the class material there were some examples. I had to look through them to understand it better.
SKN		M212: I wrote the whole program but I couldn't see the output.	HEG		M312: Oh! (surprised) Ok, so basically what you do is… oh... I think there's some instructions in Laulima. I haven't seen them though, but there are some instructions in there. You basically have to compile it and then you write a little HTML. Ahmmm, so you have to compile and write a little HTML and then put them on the directory of public_html. Here, let me show you where to put it in
SKN		M212: so I didn't know how it looked.			
			RID		M312: I will close this and then this is my listing, now I'm gonna change it. I will go back out. Let me show you my list. As you can see right here I have a bunch of folders in there and one of them is… one of the directories is ammm… public_html. That's one of the directories and so I'm gonna change it to there and then I 'm gonna list everything in here and as you can see I have slotMachine.class, slotMachine.java
SKN		M212: oh no I know all that	FBS		M312: oh ok. Then, why couldn't you see it?
ACK		M212: lol			
TLR		M212: but for some reason I still cant see it			
HYP		M212: might be my			

PRI	SEC	Chat	PRI	SEC	Talk
		crowser			
HYP		M212: browser			
HYP		M212: or something			
FBG		K411: Did you enter the code to let people access the class?	SDR		M312: mmm Oh no, no, no, no! I know what it is! You have to set the settings
ACK		M212: yah	AGR	TLR	M312: yeah, you have to
FBG		M212: chmod 644			
ACK		M212: right	SKN		M312: oh man! For me I give them more rights. I haven't tried 644, but for me I give them a lot more. It works but since 644 was mentioned, 644 might work.
TLR		M; 644			
TLR		M212: hmmm			
SKN		M212: yah, that's what I did			
SKN		M212: but it didn't work	RCW		M312: let me see how much I gave them…
			FBS		M312: Hey K, does yours work?
			HYP		M312: …you are not grabbing the java, you are grabbing the class, I believe
FBG	TLR	K411: Make sure you do it to the class, I did it to the java and couldn't figure out why it wouldn't work			
NTA		K411: mine works			
			HYP		M312:[*] the HTML, you are not grabbing the java you are grabbing the class, I believe
ACK		M212: yah I know			
FBS		M212: what browsers u guys using	FBG		M312: I tried it in firefox and IE8
ACK		K411: Same			
FBS		M212: and works?	FBG		M312: both of them work

PRI	SEC	Chat	PRI	SEC	Talk
TLR		M212: dam idk then	FBG		M312: my IE is version….8.0600118975
			RID		M312: Why don't you go to mines one if it works.
			HYP		M312: Maybe its not your
TLR		M212: dam idk then	RID		M312: can you go to that one? Can you try?
		M312: (sends link to slot machine applet)	ADE		M312: can you try now? Check if it works for you..
			TLR		M312: never mind you said it was working earlier.
SDR		M212: dam yah it works on mine			
TLR		M212: I messed up somewhere down the line			
SI		M212: lol	TLR	HYP	M312: I remember at some point it wasn't working, maybe its just your codes in the HTML part
ACK		M212: maybe			
			HES		M312: you wanna…. Wait!... can you give me the link of the website that you put it in?
			HES		M312: or if you give us the link we can probably look at it.
SKN		M212: I don't even know how to do that either			
SKN		M212: shoot im so lost in this one	EMS	FBS	M312: oh that's all right. You uploaded to a website, right?
SKN		M212: I really don't understand			
EMS		M212: my bad			
FBG		M212: oh no	FBS		M312: Oh, no, not uploaded to a website, but.. you put it in the public_html, right?
SI		M212: lol			

PRI	SEC	Chat	PRI	SEC	Talk
ACK		M212: yah	FBS		M312: and the website you were using is it the Hawaii.edu website?
		M212:?	TLR		M312: let me see if she's given us the link to
HES		M212: what do you mean by that?	HEG		M312: your name is M, I don't' see your drawing on the …. On the… I think we have a small amount of web space at Hawaii.edu and that's where a lot of the students have been uploading their applets to. Lets see…
			FBS		M312: What about the drawing, your paint applet? Did that work?
ACK		M212: no			
			TLR		M312: cuz if you put it into another website I don't know if they can grab the file from the public_html.
			ME	IA	M312: ok [*] what we are going to do is lets try to answer all the questions, lets try finishing first and then we are going to help you out. So that we can get it over with the questions and the quizzes.
ACK		M212: yah			
AGR		M212: I was thinking that			
NTA		M212: wait brb bathroom			
			FBS		M312: K, have you started the quiz?
ACK		K411: I have it open	ADE		M312: ok, lets start answering it then
ACK		K411: Okay	ACK		M312: okay
ME		K411: Does the code compile? If not, is this due to only a couple syntax errors you can	RCW		M312: I don't think there's a wrong or right answer. It is just your opinion, I think.

PRI	SEC	Chat	PRI	SEC	Talk
		easily correct? Or does the program suffer severe syntax problems?			
			EXP		M312: Cuz I compiled it many times and no errors at all
NTA		K411: Gonna wait for M	ACK		M312: oh, okay
TLR		K411: If it works, I imagine it compiles			
TLR		K411: I was too lazy to double-check it	ME		M312: yeah, I can show it to you guys as soon as M comes back
ACK		M212: ok			
FBS		IM: Any questions?	NTA		M312: so the first question was: Does it compile? If not, is this due to only a couple syntax errors you can easily correct? So what I am going to do is that I'm going to compile it. So, I'm gonna get out of this, then…
ACK		M212: no	FBG		M312: we are on number one, we are on number one of the quiz.
			FBG		M312: so we are just going down the line
			NTA		M312: so in the question was "did it compile?". So I am going to compile it for you guys right now.
					M312: (mumbling compiling commands)
NTA		IM: Ok. I will see you later on			
ACK		M212: ok			
			EXP		M312: no syntax error or everything.
			FBS		M312: ah… what is the next question?
TLR		K411: does it work and is it complete?			
TLR		K411: which it does			

PRI	SEC	Chat	PRI	SEC	Talk
TLR		K411: and is			
			SKN	SI	M312: yeah, I played with it multiple times. I mean, after I played with this, I don't think that I will play slot machines in real life.
FBS		K411: Errors? Does it crash?	FBG		M312: doesn't look like it crash or anything like that to me.
ACK		M212: no			
FBS		K411: Any parts of the code particularly good?	FBS		M312: Do you guys want me to go back to the code?
FBG		M212: the whole thing			
SI		M212: lol			
FBG		M212: cuz it works	EMS		M312: (laughing) the whole thing because it works?
FBG		K411: I thought it was good			
FBS		K411: on a scale of 1 to 5 for the next couple of questions			
FBS		K411: 5 is the best, 1 is the worst			
FBS		K411: the code was easy to read			
			TLR	RCW	M312: oh yes! I just remembered I think you two have to evaluate this code, so I can't give any answers.
ACK		K411: That's fine	ME		M312: but let me know if you want me to do anything with my code.
ACK		K411: okay			
RID		K411: so, M. I'll just throw up the questions and you can type your answer, 1 to 5			
ACK		M212: ok			
NTA		K411: I 'll average it with mine and enter that			

PRI	SEC	Chat	PRI	SEC	Talk
FBS	RID	K411: So, easy to read?			
FBG		M212: 5			
FBS	RID	K411: variable names were descriptive and relevant?			
FBG		M212: 5			
FBS	RID	K411: properly indented			
FBG		M212: 5			
FBS	RID	K411: comments were clear and helpful			
FBG		M212:5	SI		M312: you do not have to keep on putting 5,5,5,5 just because I can see it
			SI		M312: that's all right
SI		M212: lol			
SI		M212: 5			
SI		M212: J			
FBS	RID	K411: Instructions to the user were clear			
FBG		M212: 5			
SI		M212: what the hell is going on			
SI		M212: lol			
FBS	RID	K Easy to enter input correctly			
			SKN		M312: These are the questions in the test, this is the area that has to be filled up and its just typing the questions in here
FBG		M212: no, I was looking at ur code at the same			
FBG		M212: time			
FBG		M212: and it was blinking	ACK		M312: oh! (laughing) ok.
FBG		M212: 5	HYP		M312: I don't know. I opened another program earlier and maybe that's what it was going on
ACK		M212: icic			

PRI	SEC	Chat	PRI	SEC	Talk
FBS	RID	K411: error messages were clear and helpful			
FBG		M212: 5			
FBS	RID	K411: program was generally easy to use			
FBG		M212: 5			
FBS	RID	K411: any extra features?			
FBG		M312: 5			
SI		M212: lol			
SI		K411: Hah			
SI		M212: 1			
SI		M212: L	SI		M312: (laughs)
FBS	RID	K411: any additional comments?			
FBG		M312: 5			
ME		K411: you can just type what you want and I'll cut and paste it into the box			
ACK		M212: no			
SI		M312: I have			
SI		M312: put			
SI		M312: Blanca is the best ICS teacher in the world			
SI		M312: lol			
SI		K411: you don't get to answer the quiz =P			
SI		M212: stop flirting with the teacher			
SI		M312: oh yeah, crap			
ME		K411: guess we are done with that then	FBS		M312: all right, so anything else… you guys want me to do anything… with the code?
FBG		K411: not off the top of my head, no			
FBG		M212: no			
FBG		M212: good			

PRI	SEC	Chat	PRI	SEC	Talk
			HEG		M312: here! Maybe this can help you out M. It is an HTML page super simple. If you can look at it up there is just basically grabbing the, the slotMachine.class
NTA		M212: ok I got that	ACK		M312: oh, ok, that's good…
			HYP		M312: oh! Maybe the program itself doesn't work
ACK		M212: mmmm			
ACK		M212: maybe			
			HEG	RID	M312: if you want me to help you out a bit you can post your…. I guess we can make you a moderator and you can show us your code and we can help you out
ACK		M212: lol nah	EMS		M312: ok, because we can actually get more points in your assignment if you turn in a better assignment
TLR		M212: My code is horrible			
ACK		M212: oh yah			
FBS		M212: wait, did she grade this already?	FBG		M312: yeah I think 70% of the grade was graded and you can get some extra points
ACK		M212: it	SKN		M312: but, just to let you know that ah… it may take up a while to help you out. It's been a while since I've looked at this code.
			FBG	RCW	M312:I think the whole grade doesn't show until later on, I'm not sure.
ACK		M212: icic	HEG		M312: let me know anyway if you want and I will try to help you.
FBG		M212: your code is so simple	FBS		M312: yeah but it works, right?
TLR		M212: mine IDK			
SI		M212: lol			

PRI	SEC	Chat	PRI	SEC	Talk
FBS	RCW	K411: So is it bad that I didn't even know we had an assignment due yesterday?	FBS		M312: So, were you able to test it?
SI		M212: lol			
ACK		K411: not yet	SDR		M312: oh, you didn't do that!
			FBS		M312: are you still able to turn it in?
NTA		K411: I gotta do it later tonight			
SKN		K411: yeah I just lose points			
SKN		M212: nah she is only gonna knock off some points			
SKN		M212: not a lot	ADE		M312: oh man! Its something new, so go for it
ACK		K411: yeah			
			SKN		M312: cuz I thought it was going to take me a couple of days so I started it last night but I finished it in an hour or two. But it was new.
FBS		M212: did yours work	FBG		M312: yeah, yeah it works, did you try yours?
SI		M212: dam			
FBG		M212: yah mine didn't			
			HEG		M312: you know, the class materials give a lot of help, you know?
SKN		M212: kept giving me an error			
FBS		K411: It is another applet thing?			
ACK		M212: yah	FBG	SKN	M312: oh no! It is not an applet, it is a simple program, so… See the thing is, I haven't seen any of the lectures I have only seen the power points and then used the class material.
ACK		M212: no			
FBG		M212: Strings			

PRI	SEC	Chat	PRI	SEC	Talk
FBG		M212: same here	SI		M312: but I had experience with Java before. It was easier for me, I bet.
SI		M212: shit	SI		M312: and I think there were times when I was getting hard times, so I think, if you didn't have that much experience and you haven't been watching the videos I think you are going to get lost for sure. Unless you are like super genus!
SI		M212: lol			
SI		K411: I like that she uses Muse as an example	SI		M312: and then ah… and then also the bad thing about
SI		M212: lol			
			SI		M312: oh yeah! Yeah! That was pretty good. I was looking for a supermassive black hole though.
SI		K411: Hah	FBS		M312: in fact… do you guys need help or anything with the new one? The thing is I can't show you the code because it is really easy to spot if it is a copy from someone else's. Like there is so many ways to write the code that it is super easy to tell if it someone elses…
FBG	TLR	K411: I should be able to figure it out	SKN		M312: yeah! I think the lectures help a lot but I haven't seen any of the videos yet. What helped me out were the quick hints that I get from the PowerPoint and Google.
TLR		K411: Just gonna have to check the lectures	ACK		M312: yeah
NTA		K411: I just download the podcast if I need to see it.			

PRI	SEC	Chat	PRI	SEC	Talk
			FBS		M312: which code, the one that was due yesterday?
FBS		IM: Any questions?			
FBS		M212: is your code in the email			
FBG		M212: no			
SI		IM: J			
SI		M212: this	FBG		M312: yeah I think it was there as an attachment so you can look at it
NTA		IM: see you in a little while			
FBG		M212: the slot machine			
NTA		K411: this code was attached	HEG		M312: so you can look at it and compare it for a bit.
ACK		M212: oh, ok			
ACK		M212: okok	ACK		M312: ok Blanca, thanks
NTA		M212: gotta check			
			HEG		M312: then you can get some hints from there and hopefully you will get higher grade, when you resubmit it.
NTA		M312: blanca	FBS		M312: Oh Blanca, I have a question
FBS		M312: fast question	NTA		M312: hey Blanca
SKN		K411: And replace the .html with the .java			
SKN		K411: and it opens a web page with the code			
ACK		M212: icic			
FBS		M312: hey Blanca, I have a fast question			
NTA		M212: she's gone	FBS	RCW	M312: oh man! I don't know if you guys know but if we don't…if the code is working and there is nothing wrong with it, do we have to resubmit it? No, right?
			ACK		M312: ok
FBG		K411: I don't think so			

PRI	SEC	Chat	PRI	SEC	Talk
			FBS		M312: all right, so anything else before we head out?
			FBS		M312: all the questions were answered, right? And the quiz…
SI		IM: hey!			
NTA		IM: I'm here			
			FBS		M312: hey Blanca! I was just wondering, Do we have to turn in or resubmit our code if its working and no errors?
			FBG		IM: What you can do is that you can resubmit it as is, you know? Because with that you will get the points because you actually resubmitted, right?
			FBG		IM: If you don't think that even adding any comments matters on your output, or the feedback was good as is, then just resubmit as is
			TLR		M312: oh, ok, I think I gotta change some things in there or add a little more things to make it better, well, thanks, thanks!
			FBS		M312: No! what if the comments are all good but then the code is not 100% good if we resubmit it, Will we still get points on it? If [*] change anything?
EMS		M212: dude it was good	ACK		IM: ok, ok! [+]
			FBS		M312: ok I was just wondering because one of the questions was: Were the instructions clear?, Well I didn't put it any instructions in there. It didn't say it in the instructions so I didn't put in any instruction… ahm… is this one of the things that is supposed to be there?

PRI	SEC	Chat	PRI	SEC	Talk
FBG		M212: we all knew what to do	FBG		IM: What you can do is that you can add in a label or you can put in the HTML that says "press the button to play". Many times just having the button labeled as set or replay [+]
SI		M312: lol			
			TLR	ACK	M312: well I don't think they had any problem playing it anyways… thanks, thanks! That was my question.
			FBS		M312: Did you guys wanted me to do anything else with the code before we go?
			FBS		M312: or that should be good
ACK		K411: nope	ME		IM: just one thing. [+]
ACK		K411: should be good			
ACK		K411: already done	SI		M312: ok. All right, I guess… thank you and have a good life. I think we are all done here
SI		M212: J			
SI		K411: See you around			
SI		M212: bye bye	SI		M312: ok guys, I'll see you next time
		Left on November 23, 2010 at 8:38 pm			

Figure D6. Session 2–Team 2 Transcript

PRI	SEC	Chat	PRI	SEC	Talk
					Instructor gives instructions on how to proceed for the session.
NTA		B321: I think everybody is here	NTA		IM: Ok Guys, I need to make mmm you guys, whomever is gonna be showing the code, I am going to make a moderator.
SI		D323: Stepping up in the world Brandon. "Moderator"	OW	NTA	IM: You guys can… ok, hold on… Let me allow for somebody else to talk.. hold on… audio mmm… Ok, we (mumble) can I am going to release the mic to see if anyone has something to tell me before I go
SI		B321: /blush			
RM		D323: I am not sure I have a mic on this computer			
EMS		D323: sorry again			
RM		IM: can you guys talk?			
			RM		B321: Hello?! Can anyone hear me?
ACK		D323: I can hear you			
ACK		P423: I can hear	ACK		B321: Cool
ACK		IM: yes			
ACK		IM: I heard 2			
ACK		IM: of you			
RM		D323: I don't have a mic that works it seems	OW	SI	IM: Ok, here is the thing guys, I have Elluminate at Leeward in BS105 and I will have to teach a class at 5:30. So the way it's going to work is that once you guys get going, I will have to go to my desk, do my work, whatever and I will come back every 30 min and move the mouse and see if you guys are right there and if you have any questions
			HEG		IM: So, if your mic doesn't work, don't worry, just type on, whatever. Its ok, hehe

SKN		Discussing for the last 30 min	NTA		IM: I will promote to moderator and you will be able to take care of this.
SKN		D323: We started early			
			FBS		IM: Remember how to share the desktop?
			FBG		B321: Actually I am not quite sure. I cannot remember.
			SDR		IM: Ok then so what I can do is that I can tell you how. Ah (surprise) there we go!
FBG		P423: tools> application sharing > share desktop	EMS		IM: Oh! There you go! P is good!
			HEG		IM: You can share your whole desktop, which I recommend you do that. Or you can just share your application which will be the terminal. Otherwise, look, in here, you have the option [+]
			HEG		IM: They are exactly the ones that I use when I grade you. So these are the submission requirements and in here it is totally about the applet [+]
			EMS		[+]… Oh wow! There you go. I can see the full desktop. Do you have any questions?
NTA		IM: If you do not have any questions I will leave you alone to work			
NTA		P423: I can see desktop.			
NTA		IM: I will check in 30 min			IM: You are Beiverfever, B… all right! Bye bye(laughter)
NTA		IM: TTYL	SI		
SI		B321: Thank you Blanca.			
IA		D323: You only have to share your code.			
EMS		D323: if you want			

SI		D323: getting dizzy seeing things twice	FBS		B321: You guys still have to fill out the… the group eval in Laulima?
			FBS		B321: all right…. Can you see it now?
ACK		P423: Yup can see			
ACK	RID	D323: yeah, try closing and just sharing the unix			
			FBS		B321: Ok, how do we do this? We go on line by line? Or you guys have any…mmm.. you just… have any questions on the code
IA		D323: lets fix the pay count			
FBG		P423: Not really, like I said I think it was pretty solid overall	NTA		B321: there
FBS		D323: P you doing the group eval or me?			
			FBS		B321: So, what you were saying… you don't need the last statement right here? (7:03)
FBG		P423: Only other real thing I could comment on was that you used a number value for your randoms without declaring a constant			
FBG		P423: You should only need two conditions	RM		D323: Can you hear me?
			RM		B321: Yeah, I can hear you.
			HEG		D323: Sweet, its working. This is [*] You do need that I think. You see a double, I think you need to delete that because I have tried doing that with my code with those parameters and it wouldn't run correctly
			HEG		D323: It needs to be an equal or equal to [*]

			HEG		D323: Cuz I have the same thing. You do not need the parenthesis really, you can get rid of the parenthesis (mumble), yeah that should still work
			EXP		B321: Yeah, I just do that so that I won't get confused.
			SKN		D323: Ah ok!, and you can make iwins++ if you want to, instead of +1 cuz is the same thing. I don't know if you know that. It doesn't really make a difference. And in (mumble)
			RID		D323: Scroll up or down… ah… go, go up.
			ADE		D323: So P, are you going to do the group results or I do the group results?
ACK		P423: Uhh, I can do it after we're done here	IA		D323: So we do mmmm together, so we can do the alterations quickly
			RM		D323: I will open it up when my computer is ready. I'm having trouble with this software, I think it doesn't have much memory on it
RM		P423: I'm hearing whirring sound in the background, it's hard for me to hear what you are saying	RM		D323: I'm sorry, let me switch down the [*]. Can you hear me now?
			RM		D323: I'm saying I need to open it up but the computer does not have that much memory so it lags pretty bad.
SI		B321: He's vacuuming at the same time	SI		D323: (laughter). Yeah Vacuum! Not good
SI		P423: Ah	RID		D323: You wanna dive down and look at the…
			TLR		D323: Yeah, it looks logical
			TLR		D323: … it is separated

			SKN		D323:Just leaving a line between the east, the center and the west you can feel the difference. Just leave a line between the blank line
			SKN		D323:And between the east, the center and then west just for reading purposes.
			NTA		Lets see if I can open the… Lets see if I can open the [+]
			IA		D323: B, I'm gonna close the… well I am gonna have to ungroup thing. I am just going to ask you the questions and you just type what the answer should be and I will just type what I think too
			ACK		B321: Yeah that's fine!
			FBS		D323: This is the question 1: Compiles? I am just writing yes and the reason is that I ran it through UNIX and it doesn't give me any errors. That's all right? P…
		P423 is non-responsive	RM		D323: Still there P? Hello?
			RM		D323: P, you are still there?
SI		B321: P is vacuuming too.	RM		D323: Am I the only one that has the mic on?
SI		P423: I'm here o_o Kinda distracted though, mom is home with groceries	ACK		B321: No, I am here too
			SI		D323: ah (mumble)
			SI		D323: Ask your mom, what's for dinner?
			IA		D323: We are just going to grade it now, together. Lets go through the questions quickly.
			SI		B321: Yeah, its Harry Potter night tonite so, it is kind of a big deal.
			SI		D323: What's that?
			SI		B321: Its Harry Potter night tonite so, it is kind of a big deal.

			ADE		D323: Oh, I see.Ok, I'm just gonna ask the questions and you just if that is right or if you have comments or…
			SKN		D323: SO I just asked question 1, yes it compiles and doesn't have any errors. Nothing much about that
FBS		P423:I'm submitting this right?	ADE		D323: Question 2. We submit one version of this, So either I submit it or you submit it. I thought we just do it together quickly. You just want to do it?
ACK		P423: I will.			
FBS		P423: for #2… the output is correct now, right?			
SKN		P423: Since the typo was fixed			
ACK		B321: Yep	FBG		D323: Is correct now, yeah. But, technically was an error on.., the counts was wrong.
TLR		B321: should work properly now	FBS		D323: This is supposed to after you fix it or be before you fix it?
			FBG		B321: Hopefully now that we have fixed it! [*] I don't know
			FBG		D323:Lets just go with after, I prefer that
			FBS		D323: So it is complete now? Yes.
			SKN		D323: The reason is that we found the error, we found where the error was
			SKN		D323: I don't know what do you want to give it? Either strongly agree, I did neutrally because it wasn't relevant to the assignment. [*]
ACK		P423: Ok			
FBS		P423: Runtime errors. Does the program crash at any time?	FBG		D323: No, no errors.
			FBG		D323: Doesn't crash at any time
ACK		P423: Ok			

FBS		P423: Did you notice any particularly good error handling? Were there any particularly clear or elegant sections of the code?	FBG		D323:Sorry B but I do not see anything “elegant”. That was just… ok that’s fine
			ACK		B321: I’m cool with that...
			SI		B321: It’s my style, you know?
			SI		B321: Not elegant.
			SI		D323:Yea I think that (mumble)
			SI		B321: Just trying to get by
			TLR		D323: Just (mumble) … just pick (mumble) The number 5. I’ll just pick number five for [*]
			CHQ	ADE	D323: No comments ah!? B?
FBS		P423: What was our answer for #4 again?	HEG		D323: You should put where you put the BorderLayout b1= new BorderLayout( )
			HES		B321: Ah?
			HEG		D323:You can put that as your main, your main grid system [*] and then label the second part as your inner grid system
			EXP		B321: [*] doesn’t do that
			TLR		B321: Wait where it says, am….m right here then, right? Oh no wait
			HEG		D323: Number four is B [*] answer
			HEG		D323: You should at least put a comment next to it saying that it is the main layout [*] you really need to start putting comments
			HEG		D323: Yeah [*] cool comments [*]
			FBS		D323: You have number four P?
FBS		P423: #5?	SKN		D323: You can put public void init( ) [*] and you have the comment
			FBS		B321:What was that? Initialize it to what?
			FBG		D323: Number 5 P… Just put number 5 [*]

FBG		P423: Yeah I thought it easy to read too	EKE		B321: My biggest problem for this one was that I was actually declaring some variables inside the.. the public void init and that is where all my troubleshooting, all my errors were coming from. After that I then declared them all outside and it ran okay after.
			AGR		D323: (interrupting) you should have been doing out in the first place
			EXP		B321: Yeah [*] and then it ran ok after, so…
			EMS		D323: [*] and then you realize that you have to declare everything outside.
			ACK		B321: yeah [*]
			ACK		D323: yeah
SKN		B321: Java is never easy to read :X	FBS	RID	D323: So number 6 ah... variable names, you. [*] name.
ACK		P423: OK	TLR		D323: What number we are on?... Number seven "Code was properly indented, so it is easy to read"…
			FBG		D323: Sorry, I think you need to indent your if statements
			FBG		D323: Every you could also indent all these [*]
			FBG		D323: So, just give it a four.
ACK		P423: K			
RID		P423: #8	FBG	FBS	D323: Number 8, [*] the number 4, you agree?
NTA		B321: P is rushing too	NTA		D323: Not rushing we are just being efficient
SI		B321: dinner isn't ready yet			
SI		B321: /time management	SI		D323: There you go! Time management!
FBS		P423: You guys have been adding comments to it, ya?	FBS		D323: Ah! Next question. What number are we on
			FBG		D323: Yeah. We'd been adding comments

			FBS		D323: What question we are on? I forgot what we finally landed on
FBG		B321: 16	FBS		D323: On question 16, right? Sweet!
FBS	RID	P423: #8 The comments and documentation were clear and helpful	ACK		D323: On 8
			FBG		D323: Number 4…
			SKN		B321: yeah, over here these are the only two I have
			ADE		D323: Just give it a four
			ADE		D323: for number eight, five for number nine.
SKN		B321: I get by on 4's	EMS		D323: Nothing wrong with fours.
			FBS		D323: so is that ok P?
ACK		P423:OK			
FBS	RID	P423: #9 "The purpose/function of the program was clear"	FBG		D323: Number nine (melodic)
			RID		D323: so actually click on play and reset will [*]
			TLR		D323: [*] so we can get confused on that
FBS		P423: 4 for that one?	FBG		D323: five, not too much [*]
ACK		P423: OK	FBG		D323: same for number ten, cuz that's just a play button and a reset button.
			FBS	RID	D323: …and for number eleven [*].
RID		P423: #10 'instructions to the user were clear"			
FBS		P423: 5 also right?	ACK		D323: Yeah.
			NTA		D323: So I'll put that in here
FBS	RID	P423: #11: It was easy to enter input correctly	FBG		D323: Yeah, number 5, five for eleven too
SKN		P423: I'd say that it was >_> kinds hard to mess up clicking a button			

			SI		D323: Yeah, you gotta be really special to be able to do it.
FBS	RID	P423: #12 Error messages were clear, helpful and explained what was wrong	FBG		D323:There was no error message, so neutral number three.
SKN		P423: I don't think there even was a way to make an error > _ >	ACK		D323: Yeah, so it is neutral.
RCW		B321: You obviously didn't see my rough draft for the slot machine	SI		D323: Yeah, thankfully! (giggling)
FBS	RID	P423: #13 The program was generally easy to use	FBG		D323: Number five too. [*] with a few buttons clicked.
FBS	RID	P423: #14 Were there any extra features beyond what was required for the assignment? If so, what were they?	FBS		D323: fourteen…no. Were there any extra features B?
FBS		P423: Don't think so, right?			
			FBG		B321: I didn't have any. Mine was, my code is pretty basic.
			FBS		D323: Plain Jane?
			FBG		B321: Yes, it was in that direction.
			FBS		D323: yeah, no extra features, just what it was required
			FBS		B321: What could you have even added to it? Like Blackjack on one side or something
			FBG		D323: You can make, like, add pictures for the colors and make it roll too
			SI		B321: Oh God!!! (laughing)
			FBG		D323: And you can make it like a pull down functionally.
FBS	RID	P423: #16 any comments?	TLR	FBG	D323: Ah there is something [*] I realized that there is something [*]Just put neutral, [*] back at you. Ah!

			FBG		D323: 16 no we [*], we fixed all the problems, the problem was the icount. The error was ah… what could count [*]
			FBG		D323: It was neat and ordered
FBG		B321: mediocre at best	FBG		D323: was structured… It was structured in a logical format
			EXP		B321: Pretty much the highlighted part right here was where the error was coming
			AGR		D323: yeah[*] in here where the error was
			FBS		D323: It doesn't work now? I mean trying to get it to work…it works?
ACK		P423: Ok I got it.	FBS		B321: did you see that? (laughing)
			FBG		D323: That's weird
			TLR		D323: Illegal start of expression. [*] the equals, [*] the plus [*]
			HYP		D323: And it should be for the other one too [*] place [*] It should work now.
			FBS		B321: Is it counting?
			SKN		D323: You can see all the [*] in it
			EXP		B321: Yeah, it was. Are the games won added to the [*].. yeah, everything seems to be fixed.
			ACK		D; Ok cool
			FBS		D323: Ok, so everything's done?
NTA		P423: The only change I would suggest, purely as an aesthetic thing	NTA		B321: Wait! One more suggestion.
			ACK		D323: All right then
FBG		P423: is to move the 'games won' label so that it's next to its text field	AGR		B321: Yeah! That is how I had it before too.

FBG		P423: and put the PLAY button on top of the RESET button	AGR		D323: Yeah, that's a good idea
			AGR		B321: yeah, I will fix that in the revision. That's a good one
FBG		P423: just would look better that way.			
			ACK		D323: ok.
FBS	NTA	IM: If you do not have any questions I will come back in 30 min.			
NTA		IM: I'm gonna teach now	NTA		D323: I think we are done Blanca.
NTA		P423: I think we're about finished actually.			
			NTA		D323: We do finish all the questions
EXP		B321: One line was messing up my code	EMS		IM: I guess you guys.. oh very well. I guess you
			SKN		D323: We met earlier too, so yeah, we did [*] had some comments…
			FBS		IM: Oh so you met a little bit before to get something done?
SKN		P423: Yeah, also fixed an error in the code that was messing up the Games Won counter	FBG		D323: No we just arrived in the room early, so we just talked about it before you came
			NTA		IM: Oh ok. Ok Well, ah, I guess you guys have no questions so I'll just leave. When you are done just leave and I will come back and turn this off.
ACK		PT Thanks ^^	ACK		D323: Ok
ACK		B321: Thank you	ACK	SI	IM: Ok so you guys take care (laughing). I will see you in… cyberspace. Bye
			SI		D323:Bye
ACK		B321: Thank you for your time everyone	SI		D323: Brandon I can see your background
SI		B321: Oh really?	SI		D323: Yeah, I guess I can

			SI		D323: yeah it is quite strange to just see the Unix screen.
			SI		B321: Oh yeah how you even do it anyway?
			SI		D323: I think you pressed the wrong thingie. I think you pressed share screen to share the whole screen instead of just share a specific program.
			SI		B321: You can see the communist flag in the background
			SI		D323: Yeah, I can see that and so your [*] on the bottom
			SI		D323: ****
SI		B321:/shame	FBS		D323: Ok, so we are done?
FBS		P423: Alright, we about done? It's about dinner time here > _ >			
			FBG	SI	D323: yeah almost hungry
NTA		P423: Thanks everyone J submitting the thingy now.	SI		D323: Good luck with ah … all the code
			SI		B321: [*]
			SI		D323: same here [*]
SI		D323: cheers P			
SI		D323: thanks for submitting that			
SI		D323: see you laters			
SI		P423: Take care			
		Left on November 18, 2010 at 5:33 PM			

Figure D8. Session 2–Team 3 Transcript

PRI	SEC	Chat	PRI	SEC	Talk
			FBS		B131: ok, does anybody have questions about my code?
					Instructor explains the procedure
			EXP		B131: [[+]] If you look at my code, I kinda took things from previous things that I wrote and just coded from there, for instance. You may see, you see this description? It shouldn't be in there but I just used it to code the dice roll program that I wrote earlier and I just used it as a shell and to start programming from there. [[+]}
			EXP		B131: We had to use 5 colors, so I generated a random number between 1 and 5. I set the colorwheel to number 6 because number 1 to 5 will be selected -1 yeah? But I did it, each button I called it a slotwheel and they have numbers 1,2 and 3 so there are 3 different slotwheels. and they will correspond to whatever colors I pick
			EXP		B131: These variables, "games won", "games played" are the ones that were required for the coding so [[+]]
			EXP		B131: [[+]] it will just randomly select the colors between 1 and 5, so I defined the colors over here. One was red, 2 was blue, 3 was green, 4 was yellow and 5 was cyan. I did not make 5 in here because if it was not 4 it had to be 5. that is the only reason why I don't have it equal 5. [[+]]
			FBS		B131: ok, any questions?
FBS		B131: any questions			

PRI	SEC	Chat	PRI	SEC	Talk
FBG	FBS	C133: none for me. Except, was there ever a time when you couldn't see all your buttons?	RM		A432: sorry, can you guys hear me?
RM		C133: (sorry I don't have a mic today)			
ACK		C133: yep	TLR	FBS	A432: Even when it was matched up with the light blue color, sometimes it wouldn't count it as a won game. So I don't know [[+]] sometimes it counted it and sometimes not. I was trying to see if it was every other one but it was inconsistent.[[+]] I don't know if anybody else noticed that or B [[+]] in your code?
					A explained that sometimes the colors matched but the applet wouldn't count the game as won
			TLR		A432: sometimes it counted it, sometimes it wont. I tried to see if it was every other one, but no. It was inconsistent.
			EXP		B131: yeah. I thought that was strange too because I cant see why it would do it. At times it would do it, sometimes it wouldn't, so I cant see the flakiness in the logic, I guess, of why it would not occur at times. It was just strange. I'm still looking at it and trying to figure out what is actually going wrong but it should be straight forward because it is a simple thing, yeah? Simple logic.
					A narrows the problem to occurring when the light blue [cyan] is involved
			ME		A432: if you have any questions,

PRI	SEC	Chat	PRI	SEC	Talk
					you can jump in at any time.
EMS		R230: Thanks, and my apologies for the delay.			
			EXP		B131:[+] basically it starts from here, if you hit the play button, it generated the slot colors over here.
			EXP		B131:I used integer numbers to generate whatever slot num * one through five for the colors, and, this is how I did it. Position 1 was red, blue, green, yellow, cyan. Cyan was supposed to be number 5. Number 1 is red, two is blue and so forth. The same colors for the second position on the wheel and of course the same goes for the third.
			SKN		B131: A made a comment that sometimes when you hit the cyan color sometimes the total would come up and sometimes the total would not accumulate.
			HYP		B131: since there is supposed to be numbers from 1 through 5 and its always cyan, I am assuming that all the numbers are fine, you know, by logic, because of the way this statement is structured. Maybe its not 5, but its an integer, so I assume its 5. So, maybe I should double-check that. That's the only weird thing but it shouldn't come out cyan if it is 1, 2, 3 or 4. So I gotta double-check that. Maybe that's where the logic goes kinda flaky, maybe that's just where this area just doesn't work right. I'll double-check that one.
			SKN		A432: yeah I saw that part. I was just wondering if it doesn't make the numbers up until four. I wonder, that may be the area. And then, I guess C133 had a question there. In the chat. If there was ever a time

PRI	SEC	Chat	PRI	SEC	Talk
					when you couldn't see all of your buttons. I didn't notice anything but B, anything?
			EXP		B131: I never had a problem with the color buttons, mm they ran. I think you run [+] I'm not using windows per se to run my program, I'm using linux actually to access the internet. There are thing isn Linux [+]
			SKN		[+] to tell you the truth I never ran this in a mac or in a windows system.
FBS		IM: Hi Everyone. Any questions?			
			FBS		B131: oh yeah, I have a question. When I generate a random number between 1 and 5, * I'm supposed to get a number between 1 and 5. I shouldn't get a zero, correct? [+]
			FBG		IM: what happens is that if you have 5 as your limit, it will generate a number between 0 and 4 and then if you want to make it between 1 and 5, you have to add 1 so that it goes… because you know that by definition Java is 0-based.
			FBS		A432: * you defined your number range?
			FBG		B131: ahm… I just picked 6 as my number range, so I thought I would be generating a number between 1 and 5. Maybe a zero is appearing over there. I have to double-check that
FBG		R230: I ran it a bunch of times in jGrasp and cyan seems to work fine.			

PRI	SEC	Chat	PRI	SEC	Talk
			RM		(A asks why the game will only run 999 times if it is not in the specifications)
			CHQ	FBS	A432: [+] I guess the number of games played, when you go pass 999 [+] Why did you stop the game at 999.
			EXP		B131: I guess that because it was not really a requirement of the program, I didn't worry about it. I was just going on with the parameters of what she asked us to do. I guess that there is not request to exceed 999 or even 9999 or you know, it is just a program to demonstrate a slot machine, a simple thing. Make sure you can add if all the colors match and make sure that the number of games played just tallied correctly that's about it, so I guess I just didn't really concern myself with how much I really put in that box because there was no requirement for it. That is the reason why I didn't really care about that part.
			RM		A432: (breaking up speech)*
			FBG		B131: it doesn't really take much to make the box bigger but I didn't wanna worry about it [+] I just wanted to get it done.
FBS	SI	R230: how did you manage 1000 games? My mouse would break.			
			HYP		B131: the only thing I would probably do is that I will probably put a trace because I want to double-check the random numbers that's coming out. My suspicion is that probably its getting a zero some place and that's the reason why it is

PRI	SEC	Chat	PRI	SEC	Talk
					not coming out when o get the cyan colors, sometimes the color would increment, sometimes it wouldn't. So I have a sneaking suspicion that its probably not a five.
			TLR		B131: what I'm gonna do is, yeah, set it to four. So, it pick four so that one to four so that I will get rid of the zero, if there is a zero coming out and if the number 4 comes out then I will add one to it so it will be five. I will fix my code a little bit later to see how it runs.
FBG			RM	FBG	(*A, breaking up sound but supports R's comment about generating the number from 0 to 4)*
			EXP		B131: yeah that is the adjustment that I just made actually in the code, let me see where I put the adjustment, I changed it to five, yeah? (he says scrolling to the top of the code and pointing to the number 5)
			TLR		That's where it is. Now I just gotta increment it a little bit down later in my code.
			EXP	FBS	B131: you guys saw the adjustment that I made, right? So, that way if there was a zero crops up, then it will be a one. If a four comes up there will be a five. So it should technically work
					A432: as for me I don't really have more questions on the code. I mean it was really easy to read and there was a lot of notes in there, documentation * I mean that really helped on reading and understanding the code. * Any questions or we can just sit and wait for Blanca to come back?

PRI	SEC	Chat	PRI	SEC	Talk
FBG		R230: Well it seems to help the cyan, but I broke the win text field	ME		
ACK		R230: I second what she said.			
ACK		C133: I'm good too			
NTA	FBS	IM: I'm back. Any questions? It sounds too quiet	SKN	RM	A explains what they are doing.
ACK		IM: ok			
ACK		IM: thanks!			
			FBG		B131: I guess we are all done that's why its quiet.
			HES		A432: we didn't do the form inspection yet. Is that something that, I guess me and C133 and R230 are supposed to do. I don't know cuz last time we were here, they were doing my code so I left the room and they finished it. So I don't know what to do from here on.

Figure D7. Session 3–Team 1 Transcript

PRI	SEC	Chat	PRI	SEC	Talk
			OW	FBS	IM: okay, what happened in the 1st session that I have, the author was not there. So I have the other 2 people work without the author. It kind of makes it okay. If the author shows late, then he can answer the questions, right? But at least you guys get going. It is late in the semester. It is hard to reschedule. You guys have finals. So, before anything, I want to know, if anybody has any questions.
			FBS		B131: yes, do we have to resubmit the CD shelf? Program again?
			FBG	IA	IM: okay, I know that you got full credit. But, even if you got full credit, please resubmit just putting in the body of the e-mail that you consider that this is good after the session and that you have nothing to change and that you are resubmitting the same thing. I just want to have the resubmission for the record. So that I can say, this person resubmitted so, I give the points.
			FBS		IM: do you guys have any other questions?
			FBG		B321: not me.
			ADE		IM: okay guys, since the author is not here, I don't want to keep you waiting since I know, that you probably have schedule this, for an hour or so… So I will take any of you that wants to show the code. So, who wants to be moderator and show the code?

PRI	SEC	Chat	PRI	SEC	Talk
			ADE		B131: have B321 do it.
			FBS		IM: okay B321, and you do it?
			TLR		B321: actually, I may have to redownload the program.
			SKN		IM: oh, that's not a problem, I can just open it in my computer and I can give you access in my computer. Would you like that?
			ACK		B321: okay
			NTA	FBS	IM: okay, it is going to take me a couple of minutes so, I can locate… Whose code is that?
ACK		B131: none			
FBG		B321: K411-last name			
FBG		B131: K411-name	RID		IM: okay, what I am going to do, is that I will share my desktop and, the whole desktop and… B321 you will take over my desktop. Let me see, okay I think you have it in you try moving around
			ACK		B321: yeah, it works!
			NTA		IM: okay, I will see you in 15 min. and then, I would come every 15 min., okay guys?
			OW	FBS	IM: if you do pico CD shelf or pico CD exception you will find K's code. So, you just have to act as if the code was there, I have already saved it. You can see his code right away if you do pico. Okay?
			FBG		B321: yes I did
			HES		B131: thank you for the questions Submit the questions? Or the review?
			RID	FBS	B131: okay, I got the review up in my computer, so I guess we can start with question 1. So, did the code compile?

PRI	SEC	Chat	PRI	SEC	Talk
FBS		B131: B, did you know and run the code?	FBG		B321: yeah.
SI		K411: hey guys!			
EMS		K411: sorry I'm late			
SI		K411: had a final that ran late.			
SI		B321: all g	FBS	IA	B321: hey K, Blanca told me to review your code. She opened your code, we were just going to review it without you, but, since you're here now, should we, I don't know, wait for her, and then have you open it or should we just keep reviewing it like this?
			FBS		B321: what do you think B131?
FBG		K411: up to you guys	ME	FBG	B131: I think that K should just review the code with us, when Blanca comes up we can just tell her what happened. I think she'll be okay with it.
FBG		K411: you can keep reviewing like this if you want	RID	FBS	B321: okay then, did the code compile?
NTA		K411: and I'll just answer questions you guys have			
NTA		K411: or point stuff out			
ACK		B321: Y			
FBG		B131: I have trouble, line 15 in CDshelf was too long, syntax error			
			AGR		B131: yeah, right.
			EXP		K411: I don't know, mine worked. Maybe my screen was bigger or, my window was opened, longer, I am not sure.

PRI	SEC	Chat	PRI	SEC	Talk
HEG		K411: did you just copy and paste it? You have to fix formatting when you copy and paste sometimes.	AGR		B131: yeah K, that could have been the problem.
			ACK		B321: it's good then.
			FBG		B131: generally, I have no problems with it.
			ACK		B321: okay
			FBS		B321: did the program crash at any time?
			FBS		IM: guys, I noticed that K is here. Would you like K to show his code as opposed to having B321 show it?
RID		B131: let K take over	RID		B321: K, can you please share your code from your computer?
			ACK		K411: sure
			NTA	FBS	IM: okay guys if you have any questions I'll be back in 10 min. okay?
			NTA		K411: I'm not really sure how to show desktop, so give me a minute to improvise
			HEG		IM: click on the icon that shows a monitor.
HEG		B131: tools >application sharing > share application	ACK		K411: got it.
			RM		K411: so, you guys can see my code, yeah?
			ACK		B321: yeah
			ACK		B131: yup
EXP		K411: so this is the CDshelf part.			
SI		B321: just kidding			
EXP		K411: and…			

PRI	SEC	Chat	PRI	SEC	Talk
TLR		K411: the rest runs the program? I don't really know what else to explain			
FBS		K411: do you guys have any questions about any part of the code?	ACK		B321: I am good
FBG		B131: no comments			
FBG		B131: in the code			
EXP		K411: and this is the CD object itself	CHQ		B131: but, what Blanca wants is actually comments. The point of explaining what you want to do, yeah? There is no comments in any of your code!
EXP		K411: no, I'm not big on comments			
SI		K411: mostly because I am lazy			
TLR		K411: but that's a bad habit, I know			
FBS		B321: what number are you on?			
ACK		B321: same here			
ACK		B321: lazy	RID		B131: that would be question 8 in the… Things that were supposed to be doing.
			ACK		B321: okay
			SI		B131: I used to be a programmer back in the 80s, I kind of quit. But, but, I would let you guys know that I had worked with a program that was 250 pages long just code itself. There was no comments in there. I promise you guys, that you would be at a loss working with that program. 260 pages, 60 lines on each page figure that out.[*]
EKE		K411: one thing I couldn't get to work with my CDshelf			

PRI	SEC	Chat	PRI	SEC	Talk
		was I couldn't get a decimal format to stick.			
EKE		K411: I plugged it in all over the place, but it wouldn't work anywhere			
FBS		K411: where would I have to drop that to get it to take.			
EXP		K411: you can see the last-place I put it was there before I gave up			
NTA		B321: MMM let me check my code.			
NTA		B321: no never mind.			
SDR		B321: I wasn't able to get it to work either			
EMS		B321:/fail			
EMS		K411: at least I wasn't the only one			
SKN	RCW	K411: also, Blanca said that I didn't check for negative numbers when she initially graded it.			
EXP		K411: but when I run my code, it doesn't print negative numbers			
FBS		K411: any idea what it is I'm missing?			
OW		IM: if you use the microphones you will go fast :–)			
			RID		B321: try scroll down.
			EXP		K411: as you can see, multiple decimal formats
			EXP		K411: from me trying it everywhere = P

PRI	SEC	Chat	PRI	SEC	Talk
			FBS		B321: so, your code didn't output negative numbers? That's what you're saying?
FBG		K411:it doesn't print negative prices or stock.	ACK		B321: that's weird.
			FBS		B321: so when the track (mumbling) finds negative numbers, does that make a difference?
HYP		K411: I don't know, I thought I might have forgotten to put in something to stop it from printing negative prices			
NTA		K411: but I checked it a couple minutes ago and it wasn't			
NTA		K411: for example!			
			CHQ		B131: K, when I ran your code, you know, where you have the output, you have dollars signs in front of the numbers, so I would assume that you didn't know that your format was working
EXP		K411: I think I just manually entered the dollar sign	AGR		B131: yeah, I see now. I just missed it. My decimal formatting I did it in CD.java I didn't do it in CDshelf.
FBS		K411: where did you put it in the CD?	FBG		B131: I can't remember. I did it just before the printing, that's all I did. In the toString.
ACK		K411: gotcha			
NTA		K411: hold on one sec, so when he said the door!			
EMS		K411: okay, sorry.			
FBG		B321: mine was inside the toString method			

PRI	SEC	Chat	PRI	SEC	Talk
EXP		K411: anyway, you can see I have the decimal format as part of the toString			
TLR		K411: or at least I thought it was.			
ACK		K411: gotcha			
HYP		K411: I think I forgot to actually apply it anywhere			
ACK		K411: thanks			
NTA		K411: success!			
ACK		K411: thanks a lot			
FBS		B321: what number are we on?			
FBG		B131: I don't know			
ME	FBS	K411: anyway, that's pretty much all my code. You guys have any more questions or do you want to finish filling out the evaluation?			
FBG		B321: finish evaluation			
			RID		B321: K, can you open your CDshelf again?
RID		K411: let me know if scroll down	FBS		B321: let's see, the direction are clear in that place where it says:please enter the information as required (reading) I think it's easy to input the information correctly as well. There's only a couple of lines. K, did you have an error message?
FBS		K411: you mean could I compile it?			
EXP		K411: I get that note			
EXP		K411: but it works.	HYP		B321: I think it means that if you put in and invalid input (mumble)

PRI	SEC	Chat	PRI	SEC	Talk
			CHQ		B321: like, what if you were to put in a number for album?
			TLR		B131: I think, that edits are really basically for the price and inStock because album and orders it really didn't care what you enter and that's part of the program actually there's no edits on the album or artist name.
AGR		K411: which makes sense.	ACK		B321: yeah
SKN		K411: seems a band can be named words, letters, or numbers	ACK		B321: okay.
			TLR		B131: and when you think about it, this program, the edits were actually in the input. So the exceptions would have never kicked out because to tell you the truth all the data should have been valid before it hit the array list. Yeah?
FBS		IM: any questions?	ACK		B321: not at the moment.
ACK		B131: none			
NTA		IM: okay, then I will come back later on.			
ACK		B321: thank you			
			HEG	RCW	B131: the only thing I can tell you guys is that you can put comments in your program so if you haven't been doing it, you basically have to start doing it because it makes it easier for the instructor to figure out what's going on in your program and if she sees your comments she could do that real easy. (Mumble) it's a problem you spend a long time trying to figure out the logic of the author and if you're working in the real world,

PRI	SEC	Chat	PRI	SEC	Talk
					your coworkers will pawn you all over your head if you don't add comments.
ACK		B321: noted thanks			
SI		B321: BTW what was that 250 page code for?	SI		B131: I was working at a bank, it was a main program driver for a DDA system. It's a checking system. So you had a whole bunch of edits, a bunch of how to handle the deposits, how to pay out the checks that you guys wrote. It was so complicated, 250 pages times 60 lines per page. It was a mess.
SI		B321: oh I'm familiar with DDA's			
SI		B321: I work in a vault.			
SI		B321: sounds rough though 250 pages.			
SI		K411: glad I don't have to work with 15,000 lines of code.			
			ME		B321: anyways, I think we should get back to the evaluation.
			TLR	ADE	B321: (mumble) the program easy to use? I didn't think there were any extra features. B131, the eval or do you want me to?

PRI	SEC	Chat	PRI	SEC	Talk
			FBG		B131: K, just a comment everybody codes different. I like the way you code. It was clear to me when I read it. Like I said, it just needs comments. It solved the problem in the scope of what Ms. Polo wanted so I have no problems with what you did because it met all the expectations of the assignment except for the comments like I said and that's what she really wants to see, probably. Bringing out the negative, (mumbling) was in the input and not in the output but I wouldn't worry about that. The data is supposed to be clean anyway. Up to the array list so, like I said, everything was supposed to be good. I.e. that you know the way it was assigned. To all us we enter the same information, it ran, no problems whatever.
SKN		K411: it's nice to hear feedback I know mine isn't just garbled gobbledegook.			
EMS		K411: so thanks for that.			
EMS		B131: you're welcome			
SI		B321: did any of you guys take the final?			
SI		K411: not yet			
SI		B131: not yet, working too much.			
SI		B321: oh I see.			
SI		B321: okay			

PRI	SEC	Chat	PRI	SEC	Talk
ME	FBS	B321: do you guys want to wrap it up? Or, any part of the eval that we missed?			
FBG		B131: can wrap up	TLR		B321: no more questions, so? I guess that's it.
FBS		IM: questions? :-)			
NTA		K411: I have a question about my code for you Blanca			
ACK		IM: okay			
NTA		K411: I just need to look something up real quick			
FBS		B321: so the resubmission is due this coming Friday?			
ACK		IM: okay			
FBG		IM: just resubmit ASAP :-)			
ACK		B321: sounds good.			
SKN		K411: on the returned assignment, you said I didn't have validations to make sure numbers aren't negative			
SKN		IM: I thought I have to send grades this week but I can't do that on Monday.			
FBS		K411: what were you looking for?			
FBS		K411: I thought you meant my code returned negative values, but I can't get it to.			
FBG		IM: in the CD class you need an if statement to check that the number is not less than 0 and if			

PRI	SEC	Chat	PRI	SEC	Talk
		it is just called the exception			
FBG		IM: I think it is all in your feedback that I will send you in a few minutes.			
ACK		K411: okay			
			OW		IM: and if after that you still have questions please e-mail me. I'm good at e-mail and another thing guys I'm gonna send your feedback I think that for B131 seems he had full credit I hardly have any feedback, you know? I will send you your feedback, and if any of you at any time or at any moment wants to see any of the sessions again you just need to e-mail me and I will send you the link to the sessions that you want to revisit. The you have any questions before I go Christmas shopping?
ACK		K411: okay			
ACK		B321: nope			
ACK		B131: none			
			OW		IM: okay guys I will be sending you your feedback and please just check it out. And if you have any questions just e-mail me right up and if it is kind of a complicated question don't forget to attach your code. Thank you. In the meantime it was really cool to see you I quote that "see you". And I hope not to see you next semester, at least not in this class. You guys take care of

PRI	SEC	Chat	PRI	SEC	Talk
					yourselves and take your final okay talk to you later.
SI		K411: see you guys around			
SI		B321: peace			

Figure D8. Session 3–Team 2 Transcript

PRI	SEC	Chat	PRI	SEC	Talk
FBS		M312: what if the code doesn't compile in the first place? Should we look through the whole code and try to understand it to answer the rest of the questions			
FBG		IM: Yes			
IA		IM: the idea is that you find the errors			
SKN		IM: that is the hard part but the one that helps you learn the most			
FBG		IM: I think his code compiles			
FBG		M312: I couldn't compile his code			
FBG		M312: had errors			
EMS		IM: Sorry, I thought it compiled			
GS		IM: you can find the errors. That will be "fun"			
FBG		A432: I got his code to compile			The instructor is explaining the procedure. Letting them know that very few people got full credit in this assignment
SI		IM: J			
HYP		A432: P you may need to compile with his other two files			
HYP		IM: yes I think that his parameters were out of order			
AGR		M312: yeah, they were in the same directories			

PRI	SEC	Chat	PRI	SEC	Talk
SKN		A432: His won't work with your CD and CDException files			
AGR		IM: the price and inStock			
RID		IM: but they all should compile with the same files			
RID		IM: that you need to point out			
OW		M312: k, what should we do if he's not in 5 mins?			
NTA		A432: Ok, I just opened a new subdirectory and dumped all his files in there. It worked.			
SKN		M312: I did the same and it wouldn't			
ACK		A432: Hmmmm…. IDK then.			
NTA		M312: I didn't compile cd.java and exception.java, only cdshelf.java			
RID		A432: try… it wouldn't hurt at this point			
ACK		M312: yeah			
OW		A432: If he doesn't show up, I will try to reschedule at 1:30 or 2pm today with him. If not Monday at 9pm. Either way I will keep you updated and confirm the new time with you.			

PRI	SEC	Chat	PRI	SEC	Talk
EMS		M312: this is not good, I have to switch around my schedules again if that's the case			
SI		M312: have to check with my workplace.			
SI		M312: family schedules and friends			
SI		IM: Hi			
GS		IM: There's something we can do, we can still do it			
NTA	FBS	A432: I'm going by your availability you sent me. What's open for you the next couple of days?			
AGR		M312: since A432 can compile it			
			GS		M312: all right, lets do this!
ACK		A432: ok	RM		A432: can you hear me?
			RM		M312: not really, there are a lot of mumbles and static
			RM		M312: let me change my headset
RM		A432: Having audio problems			
			FBS		A432: did you even get to look at the code?
			FBG		M312: yes, I did I had to do it to answer the personal questions at the beginning, before the session
			ACK		A432: ah, ok.
			FBS		A432: you never got it to compile on your side?

PRI	SEC	Chat	PRI	SEC	Talk
			FBG		M312: that's correct. I never got it to compile. I put it on a subdirectory, I put all of the files in there and tried to compile the CDshelf.java and it didn't compile.
			HYP		A432: I wonder if you have to compile the other stuff first.
			ACK		M312: I can try to do that now.
			IA		M312: so you have to compile the other stuff first
			NTA		A432: I don't know *, to make it work
			SKN		M312: ah ok. I looked at the code but I didn't look at what was causing the error. I didn't have that much time
			FBS		M312: were you able to compile this one?
			FBG	TLR	A432: I think so but I have no, no, no, I don't think so, I have my * in *.
			SKN		M312: ah ok…I was able to compile mine but the exceptions wasn't working with mine. I don't know what's the reason why.
			FBS		M312: Do you want me to try to compile the two other classes first on my end and see if it works?
			ACK	ADE	A432: yeah and then go back and recompile the CD shelf again.
			NTA	ADE	M312: ok, I'll try it now. I'm gonna do it in mine or should I do it in yours?
			FBS		A432: Is it working?
			NTA		M312: I'm in the process…

PRI	SEC	Chat	PRI	SEC	Talk
			FBG		M312: *, I am having an error on… java, oh no, CDShelf.java unchecked or unsafe operations
			GS		A432: Oh! Its * to have that. Its fine. Its gonna happen, you can still do it. The xlint or something?
			FBG		M312: Yes I tried that too. I'll try it again.
			FBS		A432: but there's no error messages right? Its just that thing about the
			TLR		M312: there were earlier, mmm. I'm gonna try it here but there were. I am going to try it again.
			SKN		M312: there are eight messages.
			SDR		M312: oh wait! * they are warnings
			RID		A432: yeah its warnings, you can still run it. So try run it now. It should be ok.
			FBG		M312: ok, it ran.
			RM		A432: are you able to go in the screen too? Cuz I have it sitting, you know where it says album.
			RM		M312: yeah, I cannot. I'm trying but I cannot.
			RM		A432: Ah, ok. I guess I can pull this up and then you can kind of see.
			TLR		A432: What I noticed is that the price, the decimal formatting is only a single decimal place. I guess on the * file I am not sure if he is supposed to format it in the CD shelf or… his formatting is on the CD file
			OW		M312: Would you think It would be better if I just go down the list of the questions? On the group inspection form cuz that's what we have to turn in anyways.

PRI	SEC	Chat	PRI	SEC	Talk
			ACK		A432: Ok
			IA		M312: and then I'm just gonna ask from there and then we can look at the rest after we answer the questions that way we don't have to worry about anything.
			ACK		A432: ah, ok.
			RID	FBS	M312: so the first one, does the code compile? I guess it does compile.
			FBS		M312: should I get answers for the both of us?
			FBG		A432: you can
			RID		M312: ok then I will ask the questions and am gonna tell you what I'm answering I am going to tell you what I put in my answer. So the question is: The code compiles? It the code, is this because of syntax errors that you can easily correct or is the program +
			NTA		M312: so it does compile and the reason for the answer, I am going to say, yes it compiles but I receive multiple warnings.
			SKN		A432: I think that warnings * its still ok but.. it's a standard warning.
			NTA		M312: ok I'm just gonna put, yes, it compiles.
			NTA		M312: multiple warnings
			AGR		A432: I think with warnings is still ok. It is a standard warning
			NTA		M312: I'm gonna put "yes it compiles"
			FBS		M312: she mentioned that?
			FBG		A432: Yeah in one of the sessions.
			ACK		M312: Oh okay
			SKN		A432: session 24

PRI	SEC	Chat	PRI	SEC	Talk
			SKN		M312: oh I haven't been watching the sessions I had been only watching the PowerPoint. though.
			EMS		A432: it is good that you have been able to do the work without watching the sessions
			SKN		M312: yeah it is weird. My assignments are all good. But I have to go through Google sometimes.
			RID	FBS	M312: Okay, number 2 question, is the goal to complete? Does the program perform as expected? Is the output correct?
			FBG		A432: For me the reading was hard. It wasn't formatted properly. Yeah, it's not showing 2 decimal places.
			FBG		M312: Okay, in the comments I am going to put it price wasn't formatted to 2 decimals.
			SKN		A432: And for that reason, when the entry is entered with two decimal places, or I guess any type of entry, because I tried it, with a dollar amount, and I try that with one decimal place, and I tried two decimal places it's just that output it's just one decimal place. No matter what you enter for the money amount.
			FBS		M312: Okay, when you say 2 decimal places, do you enter 00? Or have you try it with more than zeros on those two decimal places? 1, 2 or 3, 4?
			FBG	NTA	A432: oh, I tried, 0, 1 and 2. Let me try...I guess 3.
			RID		M312: I meant, for the price, 25 and $.39? Can you try that one?

PRI	SEC	Chat	PRI	SEC	Talk
			RM		A432: Okay, it comes out to… do you see on the screen?
			SKN		M312: yes, it seems, that when he prints it out, it is formatted as double and, not double doesn't change.
			SKN	FBS	M312: And that double, is not formatted. That's how it sounds like to me. Do you agree with that?
			TLR		A432: I don't understand, because when using 0, it only shows one zero but, when you use anything else but 0 it shows up okay.
			SKN		M312: Yes, because those extra zeros is not important, for the value, like in $25 with…mmm… let's see mmm how can I explain this? Those extra zeros in the end are not important, it doesn't need it.
FBS		IM: Any questions?	SKN		A432: Okay, but you see, with the output of the price on the screen, its gonna show 25.0 it will not like 00 cents.[+]
			SKN		M312: Yes, that is because he didn't put in the code for decimal formatting. So that's why I answer, in the comments.
			FBS		A432: You know, the decimal formatting? It cannot be in the CDshelf side?
			FBG		M312: I don't know, I don't remember. Oh, no, no, no! It's whatever that print will be. I will open up his code right now.
			FBG		M312: It's whenever that system.out.println are . You can either put it under CD shelf or on CD.java.

PRI	SEC	Chat	PRI	SEC	Talk
			NTA		M312: Let me, I will open up use code and we'll see. And let me see... here under CDshelf…
			HYP		M312: Okay, let's see. Okay, it's supposed to be in the CD... Either way you can put it in anyone of them, I think so but I think, that if you put it into the CD.java it could just pulled that out. You can put some code in the CDshelf.java.
			FBS		A432: Okay, but how would you fix that then? For the money?
			NTA		M312: For the money? It is just formatting code. Shoots! I don't have the line right now. I can search it now. I can search it at Google.
			FBS		M312: have you done the decimal formatting in your code?
			FBG		A432: yes. Looking for mine.
			RID	SKN	M312: I found something here. You can try something like this. (Looking into his code)
			RID	SKN	M312: this is something fast I found in Google. So, you can try something like this. It's on the Chat.
SKN		M312: String howmuch = Currency.format(amount);			
SKN		System.out.println (each gets "+ howmuch);			
			SDR		A432: oh!
			SKN		M312: it's something like that.
			SKN		B: you could put all of this into two lines, and that's basically how it should be. Let's see, okay? Should we go into the next

PRI	SEC	Chat	PRI	SEC	Talk
					question?
			ACK		A432: okay.
			SKN	SDR	M312: okay then, the prize wasn't formatted to 2 decimal places. So, it seems like the program is complete, he has CD.java and exception.java. Yeah, it's complete, no errors. Oh wait!
			IA		A432: we can do a Plan B
			FBS		M312: complete, single error found. The price wasn't formatted to two decimal places. Are you answering it too?
			ACK		A432: no! I'm just +
			RID	FBS	M312: the 3rd one is, runtime errors? Does the program crash at any time? Please describe the input or situation.
			FBS		M312: I mean, it doesn't crash, Right? Even if you put in wrong information.
			FBG		A432: it does what it is supposed to do with the information.
			RID		M312: can you show it to me?
			ACK		A432: a okay.
			NTA		A432: yes, it will say “invalid entry”.
			SKN		A432: because, in front of “in stock” I put whatever and something else and then underneath I guess, shows that it added elements to the array you know the valid way.
			RID		M312: yeah, can you try do it with a price with letters?

PRI	SEC	Chat	PRI	SEC	Talk
			FBS		A432: the price letters.
			ACK		M312: yeah.
			NTA		A432: and then, let me put Numbers in here (the in stock)
			FBS		A432: maybe, we should try to do the album with numbers yeah?
			FBG		M312: yeah. And then, the next one, just enter, enter, enter. It should close of whole program. I mean, is it supposed to close the whole program when you do that? Can you press enter on that? Then…?
			FBS		A432: MMM...did you see what I just did? I thought, it's switched it around *(both speaking at the same time)*
			TLR		A432: if I enter 123 at the album name and then I entered the artist asked GG the price 45 and in stock 24 it would work. But if I [*] oh that's weird I never noticed that.
			AGR		M312: yeah me too
			SKN		A432: it's not [*] when I use numbers. For the album name.
			RID		M312: can you try letters again to see if it switches?
			FBS	NTA	A432: so I'm just going to do it the other way right? So, the album is going to be letters and I'll do the artist 1234 and then just a regular price
			TLR		A432: so, it switched around. That is weird. When it is regular it comes out okay. You know, when there is both alphabet characters
			TLR		A432: this is weird

PRI	SEC	Chat	PRI	SEC	Talk
			TLR		M312: yeah, if you scrolled in the CD shelf, I mean in the CD.
			ACK		A432: I see.
			TLR		M312: it's just switched around. I get a little confuse by that too. I noticed it in the end. It was kind of confusing.
			ACK		A432: whatever!
			ME		M312: so this is number 3. So, this is answer.
			HYP		A432: actually, I think, for the correctness, I think, that's because of what we just found, we have to go back to question 2. As another error. And then number 3, we can just do it like there wasn't any*.
			NTA	TLR	M312: so, I am going to put input in artist an album are switched on the output (A talks at the same time), okay I completed it. [*]
			NTA		A432: changing number 2 to c. That means C not B.
			NTA	FBS	M312: okay, I change it there. Now, number 3. Runtime errors, did the program crash? I put no errors.
			FBG		A432: yeah, no errors.
			NTA		M312: and then, in the reasons for that, I will just put that it didn't crash.
			ACK		A432: yeah.
			RID	FBS	M312: elegance? Did you notice any particularly elegant error handling?
			FBG		M312: it was okay for me.
			ACK		A432: yeah.
			FBG		M312: yeah, sort of okay. And then, the code was easy to read.

PRI	SEC	Chat	PRI	SEC	Talk
			FBG		M312: I mean, I don't mind, it's space out, it's decent for me
FBS		I M: any questions?	TLR		A432: are we neutral? I don't know, if you want to give them…
			FBS		M312: like 4? a little better. I'm not sure. So you are [*]*(laughing)*.
			FBG		A432: *(mumbling)*
			FBS		M312: is that a no?
			FBG		A432: yeah yeah, neutral number.
			RID	FBS	M312: were the variable names descriptive and relevant?
			FBG		A432: no. *(Mumbling an explanation)*
			TLR	RCW	M312: if I look at the variables, I can see, why he named them that way so like, S1obj, I think it means string 1 object. But that has really nothing to do with that variable. So I would say one or 2.
			TLR	RCW	A432: I would say… I would vote for 2. Because, if we have to go back and review it and we've even know what it was then it's kind of hard to figure it out without totally examining the whole code.
			SKN	FBS	M312: yes, yes, harder. Then number 7 code was properly indented. It was easy to match up braces. I've seen better formatting, better spaces, but it was all right. For me, anyway, I think I would give it a 4. What would you give it?
			FBG		A432: yeah, 4.
			ACK		M312: okay.

PRI	SEC	Chat	PRI	SEC	Talk
			RID	FBS	M312: and then, number 8. For the following statements, no no, the comments and documentation where clear and helpful. I don't see comments. Or documentation.
			FBG		A432: yeah, there wasn't really any.
			TLR		M312: I'm thinking one or 2
			TLR	RCW	A432: I guess one, because, I will say one because there wasn't really anything that could have helped me.
			TLR	RCW	M312: I was thinking of 2 because of the errors that he gave. Earlier, but 1 is yeah just one.
			FBS		A432: that doesn't mean that the comments and documentation*in the code?
			FBG		M312: I'm not completely sure. That's why I was calling the errors too
			AGR		A432: yeah number 2
			ME		M312: I'm just going to ask Blanca. Questions, no thank you. We'll let you know. All right, so number 9, the purpose of the program was clear. No, just looking at this so like a quick glance is not telling me anything because no comments or anything like that if I look through it and I spend like quite a while on it then I could probably figure it out but what would you think of that? The purpose of the program was clear.
			FBG		A432: I don't know, I say one or 2

PRI	SEC	Chat	PRI	SEC	Talk
			TLR	RCW	M312: okay yeah you think it was good enough? Because I think if we spend a little time looking at it we could figure it out. Is not something but I would argue about.
			FBG		A432: 2 in that.
			FBG		M312: yeah, I'm not so happy with that so I will put one.
			TLR	RCW	A432: I guess because she says that if you are a good programmer you will have documentation in place so that when somebody else sees your code, they can just look at it and they should be able to understand they do not have the kind of dissect it out, you know?
			AGR	FBS	M312: okay, I agree with you on that. Okay and the number 10 instructions to the user were clear. Should we count the album, artist, prize and in stock as instructions?
			FBG		A432: it was*I mean, it could have given a little bit more help to the user I don't know, I will go with like 3.
			AGR		M312: yes, I was going to say 3 or 4. But yeah okay 3 it is and then it was easy to enter input correctly. I mean yes it was easy. You put input and then press enter.
			COM		A432: yeah, I think I am a 5.
			ACK		M312: yeah
			SI		A432: yeah, it was pretty easy just to enter whatever. (Laughter)

PRI	SEC	Chat	PRI	SEC	Talk
			RID	FBS	M312: yeah. That make sense okay, number 12: error messages where clear and explained what was going on. I don't know if they explained it. (Mumble)
			FBG		A432: it just said invalid entry
			FBG	RCW	M312: yeah, there was an error message, but it didn't explain it. I'm thinking, it wasn't clear I mean, invalid entry, we know that the entry was invalid, but we don't know why so, I don't know if that was helpful. And if you can explain much more about the input. I'm thinking 2, 3 or 4, I'm not sure.
			FBG		A432: we have to go with 2. It wasn't clear and it wasn't good in any way. I mean the error message was there but it didn't really help much.
			ACK		M312: yeah
			RCW	FBS	M312: yeah, I mean, it's obvious, invalid entry, that there is an error. Okay it is. Alright and then, the program was generally easy to use. It's pretty simple program it's easy. I'm thinking more of a 4. What do you think?
			AGR		A432: MMM… Yeah for his good.
			RID	FBS	M312: okay, and then, 14, were there any extra features besides those required by the assignment? And what were they.
			FBG		M312: I didn't see any extra features. Oh! What would you call the printing out what was added to the array extra features?
			FBS		A432: yeah, I was trying to, look at the code 51 that came from can we see where did that came

PRI	SEC	Chat	PRI	SEC	Talk
					from?
			NTA		M312: to make it faster I will look for it online. I am going to@theCDshelf.java.
			ACK		A432: okay.
			NTA		M312: okay, and if I cannot find it in here I will look at the CD.java.
			ACK		A432: okay.
			TLR		M312: okay, (mumbling) it could be the AL. It could be that, yeah
			TLR		A432: or yeah, I guess that's an extra feature.
			TLR		M312: I think, it counts more as a checker for him. Then a feature. So, I don't think that is a feature.
			ACK		A432: A okay,
			NTA		M312: I'm just going to make it short and simple, no extra feature.
			ACK		A432: okay.
			RID	FBS	M312: number 15. Answer only if there were any extra features. I'm not going to answer that. Because there weren't any.
			ACK		A432: okay.
			FBG		M312: any other comments about the code? Let's see… What can we comment about the code? I mean, I guess I can put the spacing made it somewhat easy to read. Do you agree on that or not?
			ACK		A432: yeah. That sounds good, yes.
			FBS		M312: do you have any other comments?
			FBG		A432: I think maybe no.*Nothing about the comments and documentation.

PRI	SEC	Chat	PRI	SEC	Talk
			FBS		M312: anything else that you see?
			FBG		A432: the variables.
			FBG		M312: short and simple, I'm going to put… Simple variables would have been helpful. Okay, other than those, that's all I can think of.
			NTA		A432: yeah, that's all.
			NTA	FBS	M312: okay, I'm going to submit it now. I'm finished. And the finished. Okay. The you wanted discuss something else about this program? Or should we just go?
			FBG	FBS	A432: I think that's it already. Do we need to wait?
			NTA		M312: it has been almost an hour.
			FBS		A432: do we need to wait for her to come back on?
			FBG		M312: I think she's back on
			NTA		M312: Blanca is typing something.
FBS		IM: all okay? Questions?	FBG		M312: we are good to go.
FBS		M312: we are good to go?			
FBG		M312: yes all ok			
FBS		IM: you are done?			
FBG		M312: yes			
EMS		IM: great!			
NTA		IM: I guess that's it then			
SI		IM: J			
SI		IM: Merry Xmas!			
SI		M312: Alright thank you Blanca and A432			
SI		A432: thanks… bye!			

PRI	SEC	Chat	PRI	SEC	Talk
SI		IM: TTYL			
SI		M312: merry Christmas all			
		Left on December 11, 2010 at 11:57 AM.			

Figure D9. Session 3–Team 3 Transcript

PRI	SEC	Chat	PRI	SEC	Talk
			FBS		IM: Do you guys have any questions for me?
FBG		D323: not yet			
FBG		P423: not yet o_o	OW		IM: okay guys then I will make a moderator so that you can explain the code. This is kind of fun because M is at the library and can talk, so he will type. In the chat. Okay?
			NTA		IM: That's what's going on.
			NTA		IM: Okay now you're a moderator so we'll be right back in about 10-15 min. and hopefully when I come back C133 will be here. You can share your code.
ACK		B213: okay			
			NTA		IM: Oh! By the way guys B was not assigned to any of the teams because he turned his code a tiny little bit late. But he's here now and it is cool because C133 is not here but be is here so

PRI	SEC	Chat	PRI	SEC	Talk
					there is three. So hopefully C133 with some and welcome B213.
			NTA		IM: I'll leave you alone. See you in a little while
SI		B213: hello Blanca			
SI		D323: Hello			
SI		P423: Howdy			
NTA		IM: I b back in 10 min.			
SI		B213: this is so last-minute			
SI		B213: LOL			
RCW		B213: luckyly I e-mailed Blanca ASAP			
FBS		D323: do you get chance to review P423's code?			
AGR		P423: better late than never eh			
FBG		B213: not even			
NTA		B213: she just told me to hop in the session			
ACK		D323: ok			
ACK		B213: thanks			
NTA		P423: ok uhhh, let me get the code copied to the screen first.			
IA		D323: so lets get started			
ACK		B213: okay			
NTA		D323: but hope to be done by 8 PM, need to eat dinner			
RM		B213: so here we go again with no sound LOL!!			
ACK		B213: yes			
NTA		D323: yeah I'm in the library			
ACK		B213: k			
FBS		D323: all okay for code sharing?			
FBG		D323: yeah			
FBG		B213: yes			

PRI	SEC	Chat	PRI	SEC	Talk
RM		P423: these are very big files actually >_>not sure how well we'll fit them on a slide			
RID		IM: just share your desktop			
FBS		P423: should I send B213 the files?			
RID		IM: bring the CDshelf out			
AGR		P423: oh right			
NTA		IM: I will send them to him now			
NTA		IM: while you work			
ADE		D323: B213, do you want to fill out the group form or me?			
FBG		B213: I don't have it???			
HES		B213: you could email it to me			
HEG		IM: it is in Laulima			
NTA		IM: just open it :-)			
ACK		B213: ok			
RM		P423: can people see my SSH window? >_>			
ACK		B213: yes			
ACK		D323: yes			
RM		B213: let me adjust my screen			
ACK		B213: awesome			
NTA		P423: the taller window is CDshelf, the shorter is CD			
RM		D323: well, I have a 14” monitor			
HEG		P423: I'm not entirely sure how these windows appear for you >_> so let me know if I should bring a window to the front.			
NTA		B213: I can see the CDshelf			
NTA		D323: I see CDshelf			
NTA		P423: just put CD on the front			
ACK		B213: okay now ic CD			
ACK		D323: ok			
NTA		IM: B213 I just mailed you the code.			
ACK		B213: thank you Blanca			
EXP		P423: these are the only two files that I changed. CD exception was unchanged from the last assignment.			
ACK		D323: ok			

PRI	SEC	Chat	PRI	SEC	Talk
RM		B213: I have too many windows open			
RM		B213: I have like 3 illuminations up LOL			
IA		D323: so B213			
FBS		D323: how do you want to do this?			
FBG		B213: I'm trying to get the list			
EXP		P423: I think the program runs smoothly for the most part, but between these two classes there is probably some extra unnecessary code			
ACK		B213: okay			
ACK		B213: good to know			
EXP		P423: mostly having to do with the error checking and exceptions			
FBG		D323: it does run really smoothly			
ACK		B213: right			
TLR		D323: I'm bad at spotting unnecessary code			
AGR		B213: same here			
NTA		B213: I'm pulling up the code now			
RM		B213: that way I can see it on my screen			
ADE		D323: B213, do you want to fill in the group evaluation?			
EXP		P423: this first IF statement here is used to see if all the form fields are left blank			
RM		D323: my PC is lagging big time			
ACK		B213: ok			
EXP		P423: if not… It goes down into further checks for valid data			
EXP		P423: but I think some of these checks are already done in the other classes			
SKN		B213: yes with the else if statements it will check to see if it's valid			
SKN		B213: and these where the checks we had in the last 2 previous classes			
SKN		B213: with if statements			
RM		D323: trying to read the de in UNIX but computer is not to happy			
EMS		D323:* code			

PRI	SEC	Chat	PRI	SEC	Talk
EXP		P423: right here in the CD class… There is a check to see if the artist is blank			
EXP		P423: there is another one for album			
ACK		B213: right			
FBS		B213 is this where its at?			
FBS		B213: where you left it blank?			
SDR		B213: oh i.c. it			
EXP		P423: but I did this to in the CDshelf class			
EXP		P423: right here			
EXP		P423: so I tried taking out this highlighted section entirely			
FBG		D323: looks like you repeat code			
EXP		P423: in the CDshelf class			
EXP		P423: but when I did that, and then tried to leave a blank artist/album name when running the code			
EKE		P423: the program crashed			
ACK		B213: oh			
FBS		B213: was there something missing?			
NTA		B213: I'm trying to look it over now			
EXP		P423: I'm not sure :/ I just know that with this highlighted bit left in the program worked.			
NTA		D323: MMMM, trying to figure it out			
RCW		P423: so after a while I had to give up and submit this as is >_>			
RID		B213: P can you highlight it for a second			
EXP		P423: I think I tried to do too much with the CDshelf class, pretty much all my data validation is done in here			
EMS		B213: I just want to look at it			
EMS		B213: thanks			
TLR		P423: I think I made the CD exception class unnecessary, LOL			
SI		B213: lol			
FBG		B213: it seemed like a duplicate work			

PRI	SEC	Chat	PRI	SEC	Talk
RCW		D323: trying to figure out what is wrong, I didn't realize this when I ran it			
FBS		IM: any questions?			
TLR		D323: can't figure out the problem			
NTA		B213: I'm just looking over both CD and CDshelf			
NTA		P423: I'm going over one of the instances where I have duplicated code			
SKN		IM: I think your code was doing the work twice, right?			
CHQ		B213: with the duplicated code it still ran…			
TLR		P423: yeah >_> then tried to do too much with the CDshelf class.			
FBG		IM: exactly, in CDshelf you only validate that you sent numbers and strings that's all			
EXP		P423: the data gets validated in CDshelf before it gets sent over to the CD class			
SKN		IM: with CD exception the ranges are validated			
SKN		IM: it is also validated that strings are more than 0 length			
SDR		P423: so all the checks that I am doing in CD and CD exception are useless at the moment, lol			
FBS		B213: so with the CDException that is the only way it will be validated?			
ACK		D323: interesting			
SKN		IM: it should be that way to keep functionality into particular classes			
SKN		IM: so if the validation rules change you only change CDException			
HES		D323: I wasn't too sure how things got read from one code to another			
			HEG		IM: I am going to talk to you because this is long.

PRI	SEC	Chat	PRI	SEC	Talk
					The instructor explains that in CDshelf you are only reading the input file CD creates the CDs by calling the constructor in the CD class. She explains the flow of information through the classes.
			FBS		IM: any questions?
FBG		D323: makes sense			
ACK		IM good :-)			
ACK		B213: yes			
EKE		P423: when I tried removing the checks in the CDshelf class though, it made the program crash, when I try to enter a blank artist or album			
SKN		B213: I was confused			
			HEG		IM: if you put a blank CD. Oh! because his CDshelf when you try to create a CD you need to use try… The instructor continues to explain that every time that a CD is created there should be a try catch. She gives an example.
			FBS		IM: did you do that Philip?
FBS		P423: but what exactly? > _ >			

PRI	SEC	Chat	PRI	SEC	Talk
RID		IM: look at my CDStore2			
FBS		B213: it is mandatory to create the object right Blanca			
FBG		IM: that one has to try catch clause every time that a CD is attempted to be created			
ACK		B213: o.k.			
FBS		B213: where are we going now?			
SI		D323: I'm amazed myUH still gets used			
NTA		IM: ok guys I will be back in 10 more min			
NTA		P423: trying to find this CDStore2			
ACK		B213 oh ok			
TLR		P423: but to be honest, I'm not sure if I have it in e-mail			
SI		B213: lol			
RM		B213: my e-mail is over its capacity			
RM		B213: so I understand			
SKN		P423: so what I think she was saying was			
HEG		B213: she mentioned object			
EXP		P423: I mean a catch (CDException) clause in my CDshelf somewhere			
ACK		B213: ok			
ACK		B213: that's right			
ACK		D323: yeah			
EXP		P423: right here is the point where the object gets created			
FBS		B213: P don't you have it right there			
FBS		B213: by the cursor			
FBS		P423: so it make a try statement here?			
NTA		P423: err, I'll save a copy of this first			
ACK		B213: k			
AGR		D323: I believe that was said			
RID		B213: then			
RID		B213: copy the same line.			
RM		D323: sorry my screen keeps freezing the time			

PRI	SEC	Chat	PRI	SEC	Talk
FBS		P423: and she was saying that in CDshelf only check to make sure that data is of the right type? And use all the class to check ranges			
FBG		D323: that is correct			
TLR		P423: so that means that this NFE down here should be removed			
RID		B213: try doing that			
FBS		B213: what was the error?			
FBS		B213: when you ran it			
FBG		P423: definitely can't remember anymore XD I just know that the program crashed			
EMS		D323: well it's it aint broke don't fix it			
EXP		P423: yeah that's what I settled on eventually			
EXP		P423: but that meant leaving in a lot of code that a new didn't have to be there >_>			
EMS		B213: more work, huh!			
FBS		P423: so this check here shouldn't be done in CDshelf either right?			
FBS		P423: if artist or album is blank			
TLR		B213: it will continue on			
TLR		B213: to the next line			
TLR		B213: the next else statement…			
FBS		B213: I might correct Philip?			
EKE		P423: I think so, but in the place of the artist/album check I need to put something else			
ACK		B213: yes			
EXP		P423: making sure that the price and stop entries are numbers			
ACK		B213: ok			
NTA		B213: I am going through my checklist			
FBS		B213: so it did compile first-time right?			
FBG		P423: Yes			
ACK		B213: okay			
CHQ		B213: reason being?			
FBS		IM: questions?			

PRI	SEC	Chat	PRI	SEC	Talk
FBG		B213: we are			
FBG		B213: figuring out a statement			
FBS		P423: reason being for what?			
FBG		B213: we have to change our code for this one…			
HYP		P423: I think right here where the cursor is, I need to put in checks to see if the price and stock values are numbers			
AGR		B213: okay that's the reason I was looking for.			
HES		P423s: my brain is fried atm >_> what statement do I use to try to change the string values into numbers?			
AGR		B213: yes price			
FBS		B213: what about dPrice?			
ACK		B213: okay you got it			
FBS		P423: so I actually do need a number format exception in this class after all, right?			
FBG		B213: yes			
FBG		B213: basically we forgot the catch statements…			
FBS		B213: what happened?			
RM		P423: not sure o_O. You disappeared and came back without me noticing			
SI		B213: lol			
ACK		P423: okay uhh			
EXP		P423: this try/catch down here…			
NTA		B213: I'm looking			
NTA		B213: at it			
FBS		P423: does it need to be moved up here?			
FBS		P423: within this statement			
FBG		B213: I believe there			
FBG		B213: up			
TLR		P423: okay I'm not sure I remember the whole nested try/catch rules			
EXP		P423: but if either price or stock don't get enter properly as numbers			
NTA		B213: I'm looking through my notes			

PRI	SEC	Chat	PRI	SEC	Talk
EXP		P423: this section won't happen at all and he just goes straight down to the NFE, right?			
FBG		B213: yes			
FBG		B213: or those that entire line goals above the else statement?			
FBG		B213: and to try and catch is in the middle			
TLR		P423: the parse statements? I thought those had to be within the try			
EXP		P423: because those statements are the risky code			
ACK		B213: yes			
AGR		B213: your right			
FBS		P423: anyway, how does this CDshelf look now?			
FBG		B213: besides longer it looks good			
SI		B213: believe me my brains were burning			
FBS		IM: you guys need anything? Questions?			
HES		P423: Mind taking at look at the CDshelf now?			
NTA		B213: we have to adjust some things			
ACK		IM: sure! :-)			
EXP		P423: I tried to get rid of most of the checks other than making sure price/stock were numbers, like you said			
NTA		B213: Philip			
RID		B213: can you scroll down			
ACK		IM: yes :-)			
RID		IM: scroll down more please			
NTA		P423: this is the bottom			
SI		B213: lol			
AGR		B213 that's what I thought			
SI		IM: :-)			
FBS		IM: you are using an array list right?			
NTA		P423: code’s actually shorter than it was before, I think.			
FBG		P423: I think so			

PRI	SEC	Chat	PRI	SEC	Talk
NTA		B213: there			
HEG		IM: then you may want to typecast when you get items out of the array list at the end			
ACK		B213: oh			
HEG		IM: just to make sure that you are getting what you think you're getting			
AGR		B213: so we do need it cast at the end			
HEG		IM: since ArrayLists allow you to add different objects usually we typecast			
HEG		IM: that is at the end of the code			
ACK		B213: ok			
HES		P423: oh… How do I do that? (wiggly smiley face)			
RID		IM: can you scroll down			
RID		IM: to the top			
RID		IM: okay here!			
HEG		IM: s= ((CD)a.get(iCount).toString( ))			
FBS		B213: what happened here?			
FBS		P423: change this line?			
HYP		IM: May be overlooking parentheses			
		The instructor proceeds to explain code errors. She's explaining how to typecast the elements that are getting out of the array list.			
		The instructor is explaining exactly what to write to make it work properly and then explaining how the typecasting and everything that was done impacts the code.			
RID		B213: can you scroll left			
RID		B213: I mean right			
EMS		B213: sorry			
ACK		B213: oh			
RID		IM: can you compile that?			
SKN		IM: but it worked before that is outstanding			
SKN		IM: if this breaks it as it was			
RCW		P423: stupid errors again >_>			
GS		IM: we are just trying to make it “very” correct			
AGR		B213: yes			

PRI	SEC	Chat	PRI	SEC	Talk
GS	NTA	IM: well Philip actually has a resubmission code now :-)			
NTA		IM: it is running :-)			
SI		B213: :-)			
ACK		IM: yes!			
EKE		P423: well, it didn't crash at least o_O but now it's still constructing even without an artist name			
SI		B213: :-)			
EKE		P423: that is something to fix and other class?			
RID		IM: let's see the part where you validate that entry to make sure it is not blank			
EXP		P423: it was in the CD class			
HEG		IM: try checking the string length after trimming instead			
HYP		IM: let's see if that works			
HEG		IM: check for a length of 0 you mean?			
HEG		IM: if (newArtist.length( ) <1)			
HEG		IM: okay let's try			
HEG		IM: but that is not validating the loop for creating elements that it's in CDshelf			
FBG		P423: seems to work now o_o			
AGR		IM: well that is good news :-)			
ADE		IM: B213 and you fill out the code inspection group?			
ACK		B213: yes I can			
NTA		IM: okay I will then e-mail you all your code feedback from me			
GS		B213: awesome job Philip			
NTA		IM: so you can resubmit			
ACK		B213: OK Blanca			
EMS		B213: thank you too			
FBG		IM: yes Philip you did a very nice job			
ACK		P423: thanks guys :-)			
ACK		B213: your welcome			
SI		IM: :-)			
SI		IM: I will see you in cyberspace			
NTA		B213: Philip I'll take care of the survey			

PRI	SEC	Chat	PRI	SEC	Talk
SI		IM: hasta la vista. Gracias very much			
ACK		B213: nah			
EMS		P423: thanks ^ ^ sorry for keeping you all so long			
SKN		B213: i learning more			
EMS		IM: it was fun for me, actually			
EMS		B213: thank you			
SKN		B213: and you know what I didn't realize			
SI		IM: :-)			
SKN		B213: the time went by so fast			
ACK		IM: great			
SI		B213: byebye			
SI		IM: ttyl			
SI		P423: take care			
SI		IM: u2			
left on December 13, 2010 at 8:29 PM					

Perl Script to Count Learning Moments

```
#!/usr/bin/perl

use strict;
use warnings;

use Data::Dumper;
use FileHandle;

my $debug = 0;
foreach my $file_name (@ARGV)
{
   my ($count_2, $count_3) = parse_file($file_name, $debug);
   print join("\t", $file_name, $count_2, $count_3)."\n";
}

exit;

sub parse_file
{
   my ($file_name, $debug) = @_;

   print "READING $file_name...\n" if $debug;
   my $file_handle = new FileHandle $file_name
      or die "Cannot open $file_name\n";

   my $file_contents = '';

   while (<$file_handle>)
   {
      $file_contents .= $_;
   }

   $file_handle->close();

   parse_file_contents( $file_contents, $debug );

}

sub parse_file_contents
{
   my ($file_contents, $debug) = @_;

   $file_contents  =~ s/[\r\n]+/\n/g;              # standardize EOL
   print "ORIGINAL CONTENT:\n".$file_contents."\n" if $debug;
```

```
    my    @lines    = split(/\n/, $file_contents);  # parse into lines
    shift @lines;                                   # remove header

    my $reduced     = "\t".join("\t", @lines)."\t"; # codes in a single line
      $reduced     =~ s/\t+/ /g;                    # use a single space between codes

    print "REDUCED CONTENT:\n".$reduced."\n"        if $debug;

    my @counts  = ();
    my @matches = ();
    my $tmp_reduced;

    @matches     = ();
    $tmp_reduced = $reduced;
    $tmp_reduced =~ s/ (FBS|HES|ADE|EKE|COM|SDR|TLR) (HEG|FBG|SKN) /push
@matches, " $1 $2 "; ' <$VAR'.scalar(@matches).'> '/eg;
    push @counts, scalar(@matches);
    print "MATCHES:\n".Dumper( @matches )          if $debug;
    print "REMAINS:\n".$tmp_reduced."\n"           if $debug;

    @matches     = ();
    $tmp_reduced = $reduced;
    $tmp_reduced =~ s/ (FBS|HES|ADE|EKE|COM|SDR|TLR) (HEG|FBG|SKN)
(ACK|AGR) /push @matches, " $1 $2 $3 "; ' <$VAR'.scalar(@matches).'> '/eg;
    push @counts, scalar(@matches);
    print "MATCHES:\n".Dumper( @matches )          if $debug;
    print "REMAINS:\n".$tmp_reduced."\n"           if $debug;

    return @counts;
}
```

Appendix E. Semester Grades and Submission Messages

Table E1. Assignment Grades

	Assignments Grades											
SBL-Online Student	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A12
A432	100.00%	100.00%	91.67%	100.00%	75.00%	97.20%	100.0%	60.0%	100.0%	100.0%	71.4%	50.0%	50.0%
B213	100.00%	100.00%	16.67%	66.67%	85.00%	43.68%	42.9%	42.9%	20.4%	96.0%	14.3%	20.0%	20.0%
C133	100.00%	100.00%	98.33%	100.00%	85.00%	98.56%	100.0%	80.0%	100.0%	100.0%	85.7%	43.8%	43.8%
K411	100.00%	86.67%	95.00%	100.00%	100.00%	97.84%	100.0%	100.0%	96.0%	100.0%	100.0%	21.3%	21.3%
M212	100.00%	50.00%	75.00%	100.00%	75.00%	86.48%	92.9%	72.0%	100.0%	80.0%	100.0%	71.3%	71.3%
M312	100.00%	100.00%	100.00%	100.00%	100.00%	75.12%	21.4%	100.0%	100.0%	80.0%	91.4%	27.5%	27.5%
B131	90.00%	100.00%	100.00%	100.00%	100.00%	99.80%	100.0%	100.0%	100.0%	84.0%	100.0%	100.0%	100.0%
B321	10.00%	100.00%	100.00%	100.00%	65.00%	57.68%	92.9%	98.4%	100.0%	96.0%	71.4%	36.3%	36.3%
P423	100.00%	100.00%	100.00%	100.00%	100.00%	84.00%	42.9%	88.0%	80.0%	100.0%	100.0%	100.0%	100.0%
D323	100.00%	100.00%	100.00%	95.00%	90.00%	92.08%	100.0%	100.0%	100.0%	100.0%	100.0%	18.8%	18.8%
Average	90.00%	93.67%	87.67%	96.17%	87.50%	83.24%	79.3%	84.1%	89.6%	93.6%	83.4%	48.9%	48.9%
Traditional	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A12
A	100.00%	100.00%	100.00%	75.00%	100.00%	85.00%	100.0%	100.0%	100.0%	96.0%	0.0%	0.0%	0.0%
B	90.00%	100.00%	100.00%	100.00%	98.67%	80.56%	100.0%	100.0%	96.0%	96.0%	94.3%	60.0%	60.0%
C	100.00%	90.00%	100.00%	75.00%	97.33%	86.11%	100.0%	100.0%	96.0%	92.0%	85.7%	50.0%	50.0%
D	80.00%	100.00%	100.00%	100.00%	100.00%	83.33%	100.0%	100.0%	96.0%	88.0%	85.7%	100.0%	100.0%
G	100.00%	100.00%	33.33%	16.67%	99.33%	85.00%	100.0%	99.0%	100.0%	100.0%	100.0%	43.3%	43.3%
H	95.00%	100.00%	100.00%	75.00%	99.33%	95.56%	100.0%	100.0%	100.0%	102.0%	100.0%	100.0%	100.0%
J	100.00%	100.00%	100.00%	66.67%	98.67%	45.00%	92.0%	0.0%	80.0%	0.0%	0.0%	0.0%	0.0%
K	90.00%	100.00%	100.00%	75.00%	100.00%	65.00%	100.0%	26.7%	100.0%	96.0%	85.7%	50.0%	50.0%
M	100.00%	100.00%	41.67%	62.50%	98.67%	55.56%	60.0%	60.0%	100.0%	40.0%	14.3%	60.0%	60.0%
N	100.00%	100.00%	100.00%	100.00%	100.00%	98.33%	100.0%	100.0%	100.0%	100.0%	100.0%	0.0%	0.0%
P	100.00%	100.00%	100.00%	100.00%	100.00%	80.00%	100.0%	100.0%	100.0%	96.0%	100.0%	50.0%	50.0%
Q	100.00%	96.67%	100.00%	100.00%	99.33%	94.44%	100.0%	91.7%	92.0%	100.0%	100.0%	100.0%	100.0%
R	90.00%	100.00%	100.00%	100.00%	100.00%	100.00%	104.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
S	90.00%	100.00%	100.00%	75.00%	98.67%	80.00%	100.0%	33.3%	100.0%	88.0%	14.3%	53.3%	53.3%
T	100.00%	100.00%	100.00%	75.00%	50.00%	50.00%	88.0%	60.0%	96.0%	48.0%	37.1%	70.0%	70.0%
Average	95.67%	99.11%	91.67%	79.72%	96.00%	78.93%	96.3%	78.0%	97.1%	82.8%	67.8%	55.8%	55.8%

Table E2. Quiz Grades

Quiz Grades														
SBL-Online	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14
Student														
A432	100.0%	60.0%	100.0%	100.0%	100.0%	50.0%	80.0%	80.0%	100.0%	60.0%	60.0%	75.0%	100.0%	40.0%
B213	80.0%	100.0%	80.0%	60.0%	100.0%	100.0%	100.0%	100.0%	25.0%	45.0%	20.0%	25.0%	20.0%	80.0%
C133	100.0%	100.0%	80.0%	0.0%	100.0%	100.0%	60.0%	60.0%	100.0%	25.0%	60.0%	0.0%	0.0%	0.0%
K411	100.0%	60.0%	100.0%	60.0%	90.0%	50.0%	80.0%	0.0%	0.0%	0.0%	0.0%	0.0%	60.0%	80.0%
M212	80.0%	60.0%	0.0%	40.0%	75.0%	100.0%	60.0%	0.0%	50.0%	30.0%	40.0%	0.0%	40.0%	0.0%
M312	80.0%	100.0%	100.0%	80.0%	100.0%	100.0%	100.0%	80.0%	75.0%	80.0%	60.0%	100.0%	100.0%	40.0%
B131	80.0%	100.0%	100.0%	80.0%	100.0%	100.0%	100.0%	80.0%	50.0%	25.0%	80.0%	75.0%	100.0%	60.0%
B321	100.0%	40.0%	80.0%	80.0%	80.0%	100.0%	80.0%	60.0%	25.0%	65.0%	80.0%	50.0%	20.0%	100.0%
P423	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	80.0%	80.0%	0.0%	80.0%	80.0%	75.0%	100.0%	40.0%
D323	80.0%	100.0%	0.0%	0.0%	80.0%	50.0%	60.0%	40.0%	50.0%	90.0%	80.0%	25.0%	100.0%	60.0%
Average	90.0%	82.0%	74.0%	60.0%	92.5%	85.0%	80.0%	58.0%	47.5%	50.0%	56.0%	42.5%	64.0%	50.0%
Traditional														
A	80.0%	60.0%	100.0%	80.0%	75.0%	100.0%	80.0%	40.0%	50.0%	15.0%	20.0%	50.0%	100.0%	80.0%
B	80.0%	80.0%	100.0%	80.0%	0.0%	50.0%	60.0%	40.0%	75.0%	80.0%	80.0%	25.0%	0.0%	100.0%
C	80.0%	60.0%	80.0%	80.0%	80.0%	100.0%	60.0%	40.0%	75.0%	25.0%	60.0%	50.0%	80.0%	80.0%
D	100.0%	60.0%	100.0%	100.0%	95.0%	50.0%	100.0%	80.0%	75.0%	20.0%	20.0%	25.0%	0.0%	60.0%
G	60.0%	0.0%	60.0%	80.0%	55.0%	100.0%	20.0%	40.0%	50.0%	100.0%	80.0%	75.0%	80.0%	100.0%
H	100.0%	100.0%	100.0%	60.0%	90.0%	50.0%	80.0%	60.0%	75.0%	40.0%	40.0%	25.0%	80.0%	80.0%
J	60.0%	60.0%	100.0%	80.0%	85.0%	50.0%	60.0%	80.0%	50.0%	40.0%	20.0%	0.0%	0.0%	0.0%
K	80.0%	100.0%	80.0%	60.0%	40.0%	100.0%	60.0%	40.0%	100.0%	25.0%	100.0%	25.0%	100.0%	40.0%
M	80.0%	100.0%	100.0%	100.0%	50.0%	100.0%	80.0%	80.0%	100.0%	80.0%	100.0%	100.0%	100.0%	100.0%
N	100.0%	100.0%	100.0%	60.0%	100.0%	100.0%	100.0%	60.0%	75.0%	50.0%	100.0%	75.0%	0.0%	60.0%
P	100.0%	60.0%	100.0%	100.0%	95.0%	50.0%	80.0%	80.0%	75.0%	60.0%	60.0%	25.0%	80.0%	80.0%
Q	60.0%	0.0%	100.0%	60.0%	100.0%	100.0%	60.0%	60.0%	50.0%	80.0%	40.0%	50.0%	80.0%	60.0%
R	100.0%	100.0%	100.0%	60.0%	100.0%	50.0%	60.0%	100.0%	50.0%	90.0%	40.0%	100.0%	0.0%	80.0%
S	60.0%	100.0%	60.0%	80.0%	90.0%	100.0%	80.0%	60.0%	100.0%	65.0%	80.0%	25.0%	40.0%	40.0%
T	40.0%	0.0%	60.0%	60.0%	75.0%	50.0%	60.0%	20.0%	75.0%	40.0%	60.0%	50.0%	80.0%	20.0%
Average	78.7%	65.3%	89.3%	76.0%	75.3%	76.7%	69.3%	58.7%	71.7%	54.0%	60.0%	46.7%	54.7%	65.3%

Table E3. Exam Grades

Exam Grades			
SBL-Online	Midterm1	Midterm2	Final Exam
Student			
A432	68.8%	71.4%	77.8%
B213	47.5%	30.4%	50.9%
C133	83.8%	82.1%	79.6%
K411	82.5%	91.1%	76.9%
M212	83.8%	75.0%	75.0%
M312	87.5%	95.5%	51.9%
B131	78.8%	50.0%	63.0%
B321	66.3%	85.5%	69.4%
P423	85.0%	100.0%	92.6%
D323	88.8%	96.4%	70.4%
Average	77.3%	77.8%	70.7%
Traditional			
A	76.3%	64.3%	61.1%
B	72.5%	98.2%	79.6%
C	68.8%	76.8%	70.4%
D	100.0%	91.1%	42.6%
G	58.1%	17.9%	61.1%
H	75.0%	75.0%	55.6%
J	85.0%	73.2%	0.0%
K	78.8%	41.1%	63.9%
M	40.0%	53.6%	44.4%
N	87.5%	53.6%	69.8%
P	61.3%	75.9%	65.7%
Q	82.5%	77.9%	79.6%
R	93.8%	89.3%	88.0%
S	60.0%	89.3%	70.4%
T	63.8%	80.2%	58.3%
Average	73.5%	70.5%	60.7%

Traditional (Control-Group) Submission Messages	
A6-BMI	A6-BMI
This assignment really gave me a tough time. I could not determine how to prevent the user from entering no numeric charcters, or just pressing enter while using the scanner class. I ended up using the buffered reader. Can you give me some ideas on the scanner problem? Have a nice weekend.	I thought I sent you this file, already. Anyway, since I had the time, I thought I would make improvements. So here is my BMI assignment. Have a fabulous weekend.
Thanks for the help it still took a little while to get things just right but I felt pretty accomplished when I did :)	
This is a revision of my assignment.	
Whew! What a challenge this was… It made me wonder what other students' outputs look like. I'd be curious to see what they've come up with since there are probably lots of differences...	The more I look at it, the more things I find to change...
I'm not completely happy with what I'm turning in..It works nice..up to a point. I just wanted to turn in something just in case I cant figure out what I did wrong.	
Your code fixed. Too long to add to Laulima. I just patched it so it is not well-formatted	Here's the 6th assignment. This is a tough one.
I am sending you my assignment 6 BMI code. I know that it compiles but I know that it doesn't work exactly correct. I am sorry that I didn't have the time to put into this but I have had a very hectic week and other extenuating circumstances, (not to make excuses). I hope you willl take this into consideration while grading my assignment. Thank you for your time and have a good weekend.	
A8-Slot Machine	A8-Slot Machine
Here is my slotmachine code. I had trouble getting the actionlistener to work when the code was with it, so I moved it up and it worked. You can find the slotmachine at the same address as before	I hade some additional thoughts and made some changes to my program. I feel that it makes it look cleaner, I hope you like it. The address is
I dont know why I cannot see the colors, but I didn't want to miss turning in on the due date. If I get a better version, I will turn that in. Your feedback on this will really help.	
My slot machine is incomplete and not working. I finally managed to get it to show up, but now I gotta head to work and I don't think I'll be able to finish it in time so I'll just submit what I can. If I can get it finished I'll submit it ASAP.	Second version. Did what I could, I got the counter working but no random number generation or colors changing. This is what I could do before the deadline, but I will finish it up tomorrow morning.
It works really well with Safari. Can't open it with Firefox for some reason.	
Thank you so much for all your last minute help. It really did help me but I was able to figure some of it out for myself with some trial and error. I have once again also attached my html file because I am not too sure if it did post on my webpage, though I am pretty sure that it did. Enjoy the rest of your weekend.	
Thank you for the extension!! I am really happy with this version	
Since the deadline was extended to today I managed to get it finished and working properly. The URL is below.	
Thanks for the additional time! My applet can be found here	
This works best with Safari. Not too well with Firefox.	
Coleen really helped me out. She showed me how use a program called eclipse which makes it easier to see your errors as well as runs the program all in the same place. Very user friendlly. Once I got it to where I wanted it I moved it to unix. The stuff I was having trouble on turned out to be alot easier than I thought. I was overthinking it. Now to study for the midterm. Here is my link to my Slot Machine. I hope you like it.	
A12-CDShelf	A12-CDShelf
Here it is, my last assignment! It worked very well for me, I hope it is acceptable and works well for you as well. Have a great week and weekend.	
First submit was wrong data, please replace it with this one. Please email me if you got this submit.	
Not complete but will try to fix and visit your office.	Sorry for so many redo but this is the final verison I believe. Have a good summer.
You're probably thinking the same thing as us - FINALLY!!! I'm not sure how I did with this one because it was not easy to figure out how to make it work. Well, I'm moving on to try to study for the final. I've got a few days to cram…	Thank you SO much, Blanca for coming online late at night to help us. We were really trying to work it out only our logic was a little mixed up. We actually knew where the zero was coming from but we though we could change it to -1 after the program. It "might" have eventually come to us later, but our brains were getting a little fried. I think that it works now that you gave us all those hints.
I feel so accomplished that I was able to figure this assignment out all by myself. especially with it being the last assignment. Thanks for all your help again.	
Finally... Thanks for the help...we needed it:)	
here is my last project!!!	

Figure E1. Assignment Submission Messages: Control Group

First SBL-Online Assignment Submission Messages		
Student	**Submission Messages**	**Resubmission Messages**
B131	*blank*	A6-BMI resubmit after elluminate session
C133	Hey Ms Polo, I messaged you about my assignment on Skype, but I'm not sure if you got it. This is what it said: My program works except when I enter an invalid.. height, my program doesn't terminate. It still lets me put in a weight, and it calculates the bmi. with an incorrect weight, it just terminates.:: But I'd rather turn it in than not. So here ya go!	This one is a LOT better. Or at least, I think so :)
A100	No issues or errors.	No issues or errors.
M212	here is my assignment 6..my output is not correct i need help thank you.	here is the revised code.
B213	*blank*	NO RESUBMISSION
R230	There isn't any need to check on the code for this one. I've been messing with it for a while and it still doesn't compile.	Please ignore BMI2 this one is my final submission.
M312	tested in many different scenerios and they all worked.	NO RESUBMISSION
D323	Hi Blanca, here is the BMI code. just having trouble with logical sequence of where to place if's, else's, and try/catch. I suppoose it will come with more practise.	Here is the resubmission of assignment #6 BMI , which i didnt realize i had to do.
B321	Prof. Blanca, I'm having a difficult time getting this code to work properly. My program compiles but I was unable to add the required "if" statements and my BMI comes out all wacky. Would I be able to go into the elluminate live session with my current .java file to learn more about what I am doing incorrectly? I'm hoping for any type of feedback as I'm sure it would help. I will keep trying to finish my code. Thank you.	NO RESUBMISSION
P423	Encountered two minor issues while creating this: 1) I noticed a very strange BMI value when I tried inputting my own height and weight, low enough that I ought to be clinically dead + probably decomposed. It turns out that the BMI formula provided on Laulima is inverted in a sense. The instructions specify it as: (height(in) * 703) / (weight(lb)^2) It should actually be: (weight(lb) * 703) / (height(in)^2) Since I'm assuming you wanted us to actually calculate real BMI values instead of some theoretical values, I used the latter formula. Hopefully this is what you intended... and if you were just trying to trick us and see if we were testing the provided formula, well, I wasn't fooled! :p 2) I was unsure as to how to cause the program to close following invalid input from the user. If it was covered in sessions 12-13, maybe I'm just being a ditz and haven't found/recalled it... In any case, I looked around online and found that the System.exit method can be used to terminate the program. If it was appropriate for us to use this method, then I guess I'm alright. If not, then I guess I'll find out the way it's supposed to be done soon! Had fun making this one... despite the reminder of how unhealthy I really am > _ >	NO RESUBMISSION
A432	I went around and around in circles on this one. I thought I had it going and then I started making changes to it and then sometimes I was on the right track, but many times it got worse. I know I'm missing some of the important components in the program. I just had some trouble trying to tie it all in together to make it work. At the least, I got it to be able to compile and run, but it is by no means, anywhere near complete.	*blank*
K411	This assignment kicked my ass.	Here is the resubmittion of my code after our group's meeting.

Figure E2. Assignment Submission Messages: SBL-Online Session 1

Second SBL -Online Assignment Submission Messages		
Student	**Submission Messages**	**Resubmission Messages**
K411	URL	URL
M312	URL tested with IE8 and FF3.6.12	resubmission. i didn't touch anything because my group said there wasn't anything to be improved on. everything worked great and they knew what they were doing. i already see the grade there so i hope it's not too late :D. but i understand if it is. thank you thank you
M212	file:///C:/Users/URL i still dont understand how to view my applet so i am pretty much shooting in the dark here..	URL
B213	*blank*	*blank*
P423	URL Once again thanks so much for your help with this!	NO RESUBMISSION
D323	So i finally got the applet to work after much fustration.I never expected to spend so much time devoted to a 5 color slot machine.Here it is: URL	Here is the resubmission of assignment #8 slot machine. I have kept the code the same , as I got to compile and run with out any gliches. URL
B321	URL EDIT: Sorry forgot to include website.	resubmit: URL
B131	The html for A8-SlotMachine URL	A8-SlotMachine resubmit. URL: URL
A432	URL Hi Blanca, I emailed you earlier today on this.	URL
R230	Well even with the extra day, I was unable to get this slot machine organized properly. As for posting this as a webpage, where is the reference for setting that up? I didn't even realize I had a web page	This round almost came out fine. For the most part it functions fine, however, when I was using the layout's I couldn't get the slot machine wheels set to the right size. Also since I used the North,Center,South BorderLayout I coudln't get the south end to stay close to the rest of the layout.
C133	This is all I have so far. :(Everytime I made a change, if my applet didn't change, I erased it so I at least had a starting point. I'm still trying to fix it though.	I have no idea what's wrong with this. I keep changing it and messing it up. At one point, the only color that wouldn't win was yellow. Now it's just winning randomly :(
	Sorry, I forgot the proper title. Also, the url is URL	

Figure E3. Assignment Submission Messages: SBL-Online Session 2

	Third SBL-Online Assignment Submission Messages	
Student	**Submission Messages**	**Resubmission Messages**
B131	*blank*	Resubmission, no change made to any of the java programs. Got full credit on first submission.
K411	This assignment was no fun. But it's done! Yay!	Here is the resubmission of my A12 codes.
B321	*blank*	Resubmission.
M212	here is my assignment 12..my output is not exactly how you wanted it to be..	no resubmission (he was not present in the code evaluation)
A432	Sorry... I really messed up on this assignment.	I feel so much better after working on this resubmit and seeing how I was able to work out a lot of the issues I had with my original submission. It looks so much better. I just got really discouraged and wasn't too motivated to complete it because I was having so much trouble on it. But I am glad I redid it. Anyway, thank you for putting up with me. This was a really tough semester for me with 5 classes, along with having so much difficulty with my other computer science class (discrete mathematics). I really learned a lot in your class and I had some fun in the process.
M312	After working on it for 7 hours, I couldn't think straight anymore. I hope I'll learn something new from Elluminate session. Keep getting mismatch or incompatible error. I think I understand the error but I don't know how to fix it.	resubmit. i made a huge improvement but still failed at it. i don't know if i should still turn it in but i'll do it anyway.
P423	I tried to make changes to the CD class, but all I ended up doing was breaking the program, and again, I think it at least already works fine as it is (I'm pretty sure that it won't create any invalid CD objects). So just settling for this for now. Might be able to make some changes to it after the elluminate session ^^	The program itself doesn't seem to run much differently than before, if at all, but the code seems a lot more "balanced" now thanks to the elluminate session. It was very helpful :)
C133	Once again, I'm completely lost. Are we reading the user's input, and then printing it out like your driver classes did? And then, I think I'm lost as to how I link up my array list with my CD object. (Besides creating a program like your previus driver classes. I'm gona keep workin on this until I figure it out, but I'd rather turn it in now. Again (not a lot to show for it), but that's because I have dummy programs with a whole bunch of random code on them!	FINALLY! Thank goodness for Elluminate! Thank you for all of your assistance this semester. Arrays are still gonna take a lot of work for me to fully understand, but hopefully I can get a B this semester :) so I can take 211 and 215 next semester
D323	I just realized assignemt #12 is due 6hrs before #11 , so i honestly never got a lot done. I understand what needs to be done and how to do the logic but just not enough time , sorry.	no resubmission
B213		*blank*

Figure E4. Assignment Submission Messages: SBL-Online Session 3

Glossary of Terms

Association of Computing Machinery (ACM)

Association of Computing Machinery (ACM) is the world's largest educational and scientific computing society. The ACM helps to the advancement of computer science by disseminating important resources and hosting a digital library to serve those in the computer profession (http://www.acm.org/).

The ACM is widely recognized as the premier membership organization for computing professionals, delivering resources that advance computing as a science and a profession; enable professional development; and promote policies and research that benefit society (http://www.acm.org/about).

Applet

A Java applet is a Java program that can run within a Java-enabled browser (http://docs.oracle.com/javase/tutorial/deployment/applet/index.html).

Computer Mediated Communication (CMC)

"CMC refers to any form of interpersonal communication that uses some form of computer technology to transmit, store, annotate, or present information that has been created by one or more participants" (Wolz et al., 1997). By this definition, the tools used in this study are part of CMC.

CPATH

CISE (Computer & Information Science & Engineering) Pathways to Revitalized Undergraduate Computing Education. CISE supports and contributes to the education of the next generation of computer scientists and engineers by supporting researchers in these areas (http://www.nsf.gov/cise/about.jsp).

CS1 Computer Science 1, as described by the ACM

Computer Science 1 (CS1) is the de-facto introductory programming course as described by the Association for Computing Machinery (http://www.acm.org/) in pages 22 to 24 of the Computing Curricula 2001 Computer Science Final Report (Chang, 2001) released on December of 2001.[3] This course is the equivalent to Information and Computer Science 1 (ICS111) as known at the University of Hawai‘i campuses statewide.

Introduction to computer science, (programming) is a prerequisite for many advanced courses in computer science, therefore, this course has been articulated within the University of Hawai‘i system. The articulation of this course has been done to comply with ACM standards. This facilitates student migration from one institution to another within the UH system and the United States. This is also the main reason why direct comparisons between CS1 courses as interpreted in this study are feasible.

Elluminate

Elluminate, was a web conferencing program developed by Elluminate Inc. Elluminate "rents out" virtual rooms or virtual spaces for schools and businesses to hold classes and meetings. Elluminate has been acquired by Blackboard Inc. (http://www.blackboard.com/Platforms/Collaborate/Overview.aspx). Elluminate Live! communication tools included integrated voice over IP and teleconferencing, public and private chat, quizzing and polling, emoticons, and a webcam tool. The software includes several visual tools including whiteboard, application sharing, and file transfer among others. The software also includes a record feature that allows the moderator to record the class for others to watch later as well as a graphing tool, breakout rooms for group work, and timer. The whiteboard supports the uploading of presentations for viewing on the whiteboard for class or meeting (http://en.wikipedia.org/wiki/Elluminate_Live).

[3] No changes or references were made to this document in the interim revision of the 2001 curricula that was made in 2008, which includes an update of the CS2001 body of knowledge plus commentary.

Elluminate was being adopted at Leeward Community College during the Fall 2010.

Elluminate is very powerful and useful to this project. Elluminate allows students to meet in a virtual room and to do team work from different physical locations. The Elluminate software also allows team meetings to be recorded with the click of a mouse. The recordings of studio sessions were used as evidence of the teamwork and were collected for further analysis.

Peer Review

For this study, peer review is defined as students analyzing other students' work, and providing feedback and comments (Hamer, Luxton-Reilly, Purchase, & Sheard, 2011).

Studio-Based Learning

Studio-Based Learning (SBL) is an "inquiry apprenticeship model that follows problem-based learning but allows a more pervasive person-centered approach" (Brocato, 2009). This approach is modeled after the architectural design studio. It is founded on the concepts of reflection and social constructivism (Hundhausen, Agrawal, Fairbrother, & Trevisan, 2010; Lynch et al., 2002). The key elements in studio-based learning are collaborative learning, interactive problem solving, public presentation and review by peers in small groups (Docherty et al., 2000; Woodley & Kamin, 2007).

In studio-based learning, learning is seen as "a process in which one constructs one's own understandings through active engagement with one's environment, especially through constructing and exploring one's own artifacts" (Hundhausen et al., 2008). In the case of a programming assignment, an artifact is the programming code generated by the student as part of an assignment.

SBLProject (CPATH)

The CPATH SBL project is a multi-institutional project for the implementation of studio-based learning pedagogies in computer science courses. This project was sponsored by

the national science foundation. This project was collaboration between the University of Hawai‘i at Mānoa, Washington State University and Auburn University. For more information please visit http://www.studiobasedlearning.org.

SBL-Online

SBL-Online is the implementation of the Studio-Based Learning pedagogy in an online environment with content and timing constraints set with the purpose of maximizing effectiveness.

SBL-Online attempts to be both, a preventive and a reactive measure to help the online student. The reactive part consists on using past student's errors to plan relevant SBL-Online studio sessions for the new students. The preventive part is about applying SBL-Online sessions to current students at key points in the curriculum where errors occur the most.

References

Abdelhakim, M. N. A., & Shirmohammadi, S. (2007). A web-based group decision support system for the selection and evaluation of educational multimedia. *Proceedings of the International Workshop on Educational Multimedia and Multimedia Education* (pp. 27–36). Augsburg, Bavaria, Germany: ACM.

Ambrosio, A. P., & Martins, S. W. (2011). What matters most when teaching CS1. *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education* (pp. 385–385). Darmstadt, Germany: ACM.

Artino, A. (n.d.). *Review of the Motivated Strategies for Learning Questionnaire*. Retrieved from http://www.eric.ed.gov/ERICWebPortal/search/permalinkPopup.jsp?accno=ED499083

Barker, L. J., McDowell, C., & Kalahar, K. (2009). Exploring factors that influence computer science introductory course students to persist in the major. *SIGCSE Bulletin*, *41*(1),153-157. doi:http://doi.acm.org/10.1145/1508865.1508923

Barland, I. (2008). Some methods for teaching functions first using Java. *Proceedings of the 46th Annual Southeast Regional Conference on XX* (pp. 256–259). Auburn, Alabama: ACM.

Barr, M., Holden, S., Phillips, D., & Greening, T. (1999). An exploration of novice programming errors in an object-oriented environment. *SIGCSE Bulletin*, *31*(4), 42–46.

Bhagyavati, K., Kurkovsky, S., & Whitehead, C. C. (2005). Using asynchronous discussions to enhance student participation in CS courses. *SIGCSE Bulletin, 37*(1), 111-115. doi:http://doi.acm.org/10.1145/1047344.1047393

Bray, B. (2012, June 9). Studio-based learning (Interview: Pat Donohue). *Rethinking Learning*. Retrieved from http://barbarabray.net/2012/06/09/studio-based-learning-interview-pat-donohue/

Brocato, K. (2009). Studio based learning: Proposing, critiquing, iterating our way to person-centeredness for better classroom management. *Theory Into Practice, 48*, 138–146. doi:10.1080/00405840902776459

Bruckman, A. (2002). The future of e-learning communitites. *Communications of the ACM*, *42*(4), 60–63.

Carbone, A., & Sheard, J. (2002). A studio-based teaching and learning model in IT: What do first year students think? *ACM SIGCSE Bulletin, 34*(3), 213-217. doi:http://doi.acm.org/10.1145/544414.544485

Carrington, D., Kim, S.-K., & Strooper, P. (2010). An experience report on using collaboration technologies for distance and on-campus learning. *Proceedings of the Twelfth Australasian Conference on Computing Education, 103*, 45–52.

Carter, A. S., & Hundhausen, C. D. (2011). A review of studio-based learning in computer science. *Journal of Computing Sciences in Colleges*, *27*(1), 105–111.

Carter, J., & Jenkins, T. (1999). Gender and programming: What's going on? *SIGCSE Bulletin*, *31*(3), 1–4.

Chandler, Daniel. (1996). Engagement with media: Shaping and being shaped. *Computer-Mediated Communication Magazine, 3*(2).

Chang, C. (2001). *Computing curricula 2001: Computer science*. Retrieved from http://www.acm.org.eres.library.manoa.hawaii.edu/education/education/curric_vols/CE-Final-Report.pdf?searchterm=computing+curricula

CPATH Project. (2010, 2013). Studio-based learning. Retrieved August 31, 2012 from http://iis.cse.eng.auburn.edu/~sbl/index.php

Crawford-Ferre, H. G., & Wiest, L. R. (2012). Effective online instruction in higher education. *Quarterly Review of Distance Education*, *13*(1), 11–14.

Curtis, D., & Lawson, M. (2001). Exploring collaborative online learning. *Journal of Asynchronous Learning Networks*, *5*(1), 21–34.

Denning, T., Kelly, M., Lindquist, D., Malani, R., Griswold, W. G., & Simon, B. (2007). Lightweight preliminary peer review: Does in-class peer review make sense? *SIGCSE Bulletin*, *39*(1), 266–270.

Docherty, M., Sutton, P., Brereton, M., & Kaplan, S. (2001). An innovative design and studio-based CS degree. *SIGCSE Bulletin*, *33*(1), 233–237.

Docherty, M., Sutton, P., Brereton, M., Kaplan, S., & Brown, A. (2000). The information environments program: A new design based IT degree. *SIGCSE Bulletin, 33*(1), 233–237. doi:http://doi.acm.org/10.1145/359369.359379

Duncan, T. G., & McKeachie, W. J. (2005). The making of the Motivated Strategies for Learning Questionnaire. *Educational Psychologist*, *40*(2), 117–128. doi:10.1207/s15326985ep4002_6

Edirisinghe, E. M. N. S. (2008). Teaching students to identify common programming errors using a game. In *Proceedings of the 9th ACM SIGITE Conference on Information Technology Education* (pp. 95–98). Cincinnati, OH: ACM.

Edwards, S. H. (2003). Rethinking computer science education from a test-first perspective. In *Companion of the 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications* (pp. 148–155). Anaheim, CA: ACM.

Estey, A., Long, J., Gooch, B., & Gooch, A. A. (2010). Investigating studio-based learning in a course on game design. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games* (pp. 64–71). Monterey, CA: ACM.

Evans, M. D. (1996). A new emphasis and pedagogy for a CS1 course. *SIGCSE Bulletin*, *28*(3), 12–16.

Ficocelli, L., & Gregg, D. (2011). Coping with Java as the core CS educational language: An evolutionary experience. In *Proceedings of the 16th Western Canadian Conference on Computing Education* (pp. 38–42). Prince George, BC, Canada: ACM.

Gannod, G. C., Burge, J. E., & Helmick, M. T. (2008). Using the inverted classroom to teach software engineering. In *Proceedings of the 30th International Conference on Software Engineering* (pp. 777–786). Leipzig, Germany: ACM.

Garner, S., Haden, P., & Robins, A. (2005). My program is correct but it doesn't run: A preliminary investigation of novice programmers' problems. *Proceedings of the 7th Australasian Conference on Computing Education*, *42*, 173–180.

Gerhinger, E., Ehresman, L., Conger, S., & Wagle, P. (2008). Reusable learning objects through peer review: The expertiza approach. *Innovate: Journal of Online Education*, *3*(5), Retrieved from http://www.innovateonline.info/

Gilb, T., & Graham, D. (1993). *Software inspection* (1st ed.). London: Addison Wesley.

Gonsalvez, C., & Atchison, M. (2000). Implementing studios for experiential learning. In *Proceedings of the Australasian Conference on Computing Education* (pp. 116-123. doi:http://doi.acm.org/10.1145/359369.359386

Gross, P., & Powers, K. (2005). Evaluating assessments of novice programming environments. In *Proceedings of the First International Workshop on Computing Education Research* (pp. 99–110). Seattle, WA: ACM.

Guzdial, M., Hohmann, L., Konneman, M., Walton, C., & Soloway, E. (1998). Supporting programming and learning-to-program with an integrated CAD and scaffolding workbench. *Interactive Learning Environments*, *6*(1), 143–179.

Hamer, J., Cutts, Q., Jackova, J., Luxton-Reilly, A., McCartney, R., Purchase, H., … Sheard, J. (2008). Contributing student pedagogy. *SIGCSE Bulletin*, *40*(4), 194–212. doi:http://doi.acm.org/10.1145/1473195.1473242

Hamer, J., Luxton-Reilly, A., Purchase, H. C., & Sheard, J. (2011). Tools for "contributing student learning." *ACM Inroads*, *2*(2), 78–91.

Han, J., & Beheshti, M. (2010). Enhancement of computer science introductory courses with Mentored Pair Programming. *Journal of Computing Sciences in Colleges*, *25*(4), 149–155.

Hanks, B., & Brandt, M. (2009). Successful and unsuccessful problem solving approaches of novice programmers. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (pp. 24–28). Chattanooga, TN: ACM.

Hanson, D., Schlosser, C. A., Maushak, N. J., Anderson, M. L., Sorenson, C., & Simonson, M. (1997). *Distance education: Review of the literature* (2nd ed.). Washington DC: Association for Educational Communications and Technology.

Harasim, L. (1996). Online education: The future. In T. Harrison & T. Stephen (Eds.), *Computer networking and scholarly communication in the 21st century university* (pp. 203–214). Albany, NY: SUNY Press.

Heeler, P., & Hardy, C. (2005). A preliminary report on the use of video technology in online courses. *Journal of Computing in Small Colleges*, *20*(4), 127–133.

Hendrix, D., Myneni, L., Narayanan, H., & Ross, M. (2010). Implementing studio-based learning in CS2. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 505–509). Milwaukee, WI: ACM.

Hertz, M. (2010). What do "CS1" and "CS2" mean?: Investigating differences in the early courses. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 199–203). Milwaukee, WI: ACM.

Holmberg, B. (1977). *Distance education*. New York: Nichols.

Hoyle, R., Harris, M., & Judd, C. (2002). *Research methods in social relations* (7th ed.). Fort Worth, TX: Wadsworth/Thompson.

Hristova, M., Misra, A., Rutter, M., & Mercuri, R. (2003). Identifying and correcting Java programming errors for introductory computer science students. *SIGCSE Bulletin, 35*(1), 153–156. doi:http://doi.acm.org/10.1145/611892.611956

Hundhausen, C., Agrawal, A., Fairbrother, D., & Trevisan, M. (2009). Integrating pedagogical code reviews into a CS 1 course: An empirical study. *SIGCSE Bulletin*, *41*(1), 291–295.

Hundhausen, C., Agrawal, A., Fairbrother, D., & Trevisan, M. (2010). Does studio-based instruction work in CS 1?: An empirical comparison with a traditional approach. In *Proceedings of the 41st ACM Technical Symposium on Computer Science education* (pp. 500–504). doi:http://doi.acm.org/10.1145/1734263.1734432

Hundhausen, C. D., Agrawal, A., & Ryan, K. (2010). The design of an online environment to support pedagogical code reviews. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 182–186). Milwaukee, WI: ACM.

Hundhausen, C. D., Agarwal, P., & Trevisan, M. (2011). Online vs. face-to-face pedagogical code reviews: An empirical comparison. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (pp. 117–122). Dallas, TX: ACM.

Hundhausen, C. D., & Brown, J. L. (2005). Personalizing and discussing algorithms within CS1 studio experiences: an observational study. In *Proceedings of the First International Workshop on Computing Education Research* (pp. 45–56). New York: ACM. doi:http://doi.acm.org/10.1145/1089786.1089791

Hundhausen, C., Fairbrother, D., & Petre, M. (2011). The "prototype walkthrough": A studio-based learning activity for human-computer interaction courses. In *Proceedings of the Seventh International Workshop on Computing Education Research* (pp. 117–124). Providence, RI: ACM.

Hundhausen, C. D., Narayanan, N. H., & Crosby, M. E. (2008). Exploring studio-based instructional models for computing education. In *Proceedings of the 39th SIGCSE*

Technical Symposium on Computer Science Education (pp. 392–396). Portland, OR: ACM. doi:http://doi.acm.org/10.1145/1352135.1352271

Jonassen, D. H. (2006). *Modeling with technology mindtools for conceptual change* (Vol. 3). Upper Saddle River, NJ: Pearson.

Jonassen, D. H., Davidson, M., Collins, M., Campbell, J., & Haag, B. B. (1995). Constructivism and computer-mediated communication in distance education. *American Journal of Distance Education*, *9*(2), 7–26.

Jonassen, D. H., & Rohrer-Murphy, L. (1999). Activity theory as a framework for designing constructivist learning theories. *Educational Technology Research and Development*, *47*(1), 61–79. doi:10.1007/BF02299477

Kaczmarczyk, L. C., Petrick, E. R., East, J. P., & Herman, G. L. (2010). Identifying student misconceptions of programming. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 107–111). Milwaukee, WI: ACM. doi:http://doi.acm.org/10.1145/1734263.1734299

Kanuka, H. (2008). Understanding e-learning technologies-in-practice through philosophies-in-practice. In T. Anderson (Ed.), *The theory and practice of online learning* (2nd ed., pp. 94–95). Edmonton, Canada: Athabasca University Press.

Khun, S. (1998). The software design studio: An exploration. *IEEE Software, 15*(2), 65-71.

Laurillard, D. (2002). *Rethinking university teaching: A conversational framework for the effective use of learning technologies* (2nd ed.). New York: RoutledgeFalmer.

Lee, Y., & Choi, J. (2011). A review of online course dropout research: Implications for practice and future research. *Educational Technology Research and Development*, *59*(5), 593–618. doi:10.1007/s11423-010-9177-y

Lewis, J., & Loftus, W. (2008). *Java software solutions: Foundations of program design* (6th ed.). Boston: Pearson/Addison Wesley.

Linn, M. C. (1996). Cognition and distance learning. *Journal of the American Society for Information Science*, *47*(11), 826–842.

Lister, B. C. (1998). Interactive distance learning: The virtual studio classroom. In *Proceedings of the Third International IEEE Conference on Multimedia, Engineering and Education*. Hong Kong.

Lister, B. C. (2001). Next generation studio: A new model for interactive learning. *Syllabus Northeast*.

Ludi, S., Natarajan, S., & Reichlmayr, T. (2005). An introductory software engineering course that facilitates active learning. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education* (pp. 302–306). St. Louis, MO: ACM.

Lynch, K., Carbone, A., Arnott, D., & Jamieson, P. (2002). A studio-based approach to teaching information technology. *Proceedings of the Seventh World Conference on Computers in Education: Australian topics, 8,* 75–79.

Lynch, K., & Markham, S. (2003). The winds of change: Students' comfort level in different learning environments. *SIGCSE Bulletin*, *35*(3), 70–73. doi:10.1145/961290.961533

McLuhan, M., & Fiore, Q. The medium is the message, 1962. Harmondsworth: Penguin.

Moore, M. G. (1989). Editorial: Three types of interaction. *American Journal of Distance Education*, *3*(2), 1–7. doi:10.1080/08923648909526659

Moore, M. G. (1990). *Contemporary issues in American distance education* (1st ed.). Oxford, UK: Pergamon Press.

Pillay, N. (2003). Developing intelligent programming tutors for novice programmers. *SIGCSE Bulletin*, *35*(2), 78–82.

Pintrich, P., Garcia, T., Smith, D., & McKeachie, W. (1993). Reliability and predictive validity of the Motivated Strategies for Learning Questionnaire (MSLQ). *Educational and Psychological Measurement, 53*(3), 801–813.

Qiu, L., Tay, W. W., & Wu, J. (2009). The impact of virtual teamwork on real-world collaboration. In *Proceedings of the International Conference on Advances in Computer Entertainment Technology* (pp. 44–51). New York: ACM. doi:10.1145/1690388.1690396

Reily, K., Finnerty, P. L., & Terveen, L. (2009). Two peers are better than one: Aggregating peer reviews for computing assignments is surprisingly accurate. In *Proceedings of the ACM 2009 International Conference on Supporting Group Work* (pp. 115–124). Sanibel Island, FL: ACM. doi:http://doi.acm.org/10.1145/1531674.1531692

Reimer, Y. J., Cennamo, K., & Douglas, S. A. (2012). Emergent themes in a UI design hybrid-studio course. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (pp. 625–630). Raleigh, NC: ACM.

Reimer, Y. J., & Douglas, S. A. (2003). Teaching HCI design with studio approach. *Computer Science Education*, *13*(3), 14.

Robins, A., Haden, P., & Garner, S. (2006). Problem distributions in a CS1 course. *Proceedings of the 8th Australian Conference on Computing Education, 52, 165–173*.

Rountree, N., Rountree, J., Robins, A., & Hannah, R. (2004). Interacting factors that predict success and failure in a CS1 course. *SIGCSE Bulletin*, *36*(4), 101–104.

Schaub, S. (2009). Teaching CS1 with web applications and test-driven development. *SIGCSE Bulletin, 41*(2), 113–117.

Schlosser, L. A., & Simonson, M. (2010). *Distance education definition and glossary of terms* (3rd ed.). Charlotte, NC: Information Age Publishing.

Schulte, C., & Bennedsen, J. (2006). What do teachers teach in introductory programming? In *Proceedings of the Second International Workshop on Computing Education Research* (pp. 17–28). Canterbury, UK: ACM.

Simon, B., Kinnunen, P., Porter, L., & Zazkis, D. (2010). Experience report: CS1 for majors with media computation. In *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education* (pp. 214–218). Bilkent, Ankara, Turkey: ACM.

Simpson, M., Burmeister, J., Boykiw, A., & Zhu, J. (2003). Successful studio-based real-world projects in IT education. *Proceedings of the Fifth Australasian Conference on Computing Education, 20*, 41–51.

Sleeman, D. (1986). The challenges of teaching computer programming. *Communications of the ACM*, *29*(9), 840–841. doi:http://doi.acm.org/10.1145/6592.214913

Spohrer, J. C., & Soloway, E. (1986). Alternatives to construct-based programming misconceptions. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 17*(4), 183–191. doi:http://doi.acm.org/10.1145/22627.22369

Stone, J. A., & Madigan, E. M. (2008). The impact of providing project choices in CS1. *SIGCSE Bulletin*, *40*(2), 65–68.

Sumner, J. (2000). Serving the system: A critical history of distance education. *Open Learning, 15*(3), 267–285. doi:10.1080/026805100750036881

Teague, D. (2009). A people-first approach to programming. *Proceedings of the Eleventh Australasian Conference on Computing Education, 95*, 171–180.

Topping, K. (1998). Peer assessment between students in colleges and universities. *Review of Educational Research*, *63*(3), 249–276.

Trivedi, A., Kar, D. C., & Patterson-McNeill, H. (2003). Automatic assignment management and peer evaluation. *Journal of Computing in Small Colleges*, *18*(4), 30–37.

Turner, S., Pérez-Quiñones, M. A., Edwards, S., & Chase, J. (2010). Peer review in CS2: conceptual learning. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 331–335). Milwaukee, WI: ACM.

Ungar, J., & White, J. (2008). Agile user centered design: Enter the design studio: A case study. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems* (pp. 2167–2178). Florence, Italy: ACM.

Wang, Y., Yijun, L. I., Collins, M., & Liu, P. (2008). Process improvement of peer code review and behavior analysis of its participants. *SIGCSE Bulletin*, *40*(1), 107–111.

Whittington, K. J., & Bills, D. P. (2004). Alternative pacing in an introductory java sequence. In *Proceedings of the 5th Conference on Information Technology Education* (pp. 118–121). Salt Lake City, UT: ACM. doi:http://doi.acm.org/10.1145/1029533.1029563

Wolz, U., Palme, J., Anderson, P., Chen, Z., Dunne, J., Karlsson, G., … Walker, H. (1997). Computer-mediated communication in collaborative educational settings (Report of the ITiCSE '97 working group on CMC in collaborative educational settings). In *Supplemental Proceedings of the Conference on Integrating Technology into Computer Science Education: Working group reports and supplemental proceedings* (pp. 51–69). Uppsala, Sweden: ACM.

Woodley, M., & Kamin, S. N. (2007). Programming studio: A course for improving programming skills in undergraduates. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education* (pp. 531–535). Covington, KT: ACM. doi:http://doi.acm.org/10.1145/1227310.1227490

Woolfolk, A. (2013). *Educational psychology* (12th ed.). Upper Saddle River, NJ: Pearson.