

STUDIO-BASED COMPUTER SUPPORTED COLLABORATIVE LEARNING

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Lakshman Sundeep Myneni

Certificate of Approval:

Dean Hendrix
Associate Professor
Computer Science and
Software Engineering

N. Hari Narayanan, Chair
Professor
Computer Science and
Software Engineering

Margaret E. Ross
Associate Professor
Educational Foundations,
Leadership and Technology

George T. Flowers
Dean
Graduate School

STUDIO-BASED COMPUTER SUPPORTED COLLABORATIVE LEARNING

Lakshman Sundeep Myneni

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama

May 9, 2009

STUDIO-BASED COMPUTER SUPPORTED COLLABORATIVE LEARNING

Lakshman Sundeep Myneni

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon request of individuals or institutions and at their expense. The author reserves all publication rights.

Signature of Author

Date of Graduation

THESIS ABSTRACT
STUDIO-BASED COMPUTER SUPPORTED COLLABORATIVE LEARNING

Lakshman Sundeep Myneni

Master of Science, May 9, 2009
(B.Tech., Nagarjuna University, 2006)

110 Typed Pages

Directed by N. Hari Narayanan

The field of computing is experiencing a significant decline in incoming majors. According to the latest reports, the main reasons for this decline are (a) computing is too often related to programming, and (b) introductory computing education is too closely tied to programming languages. The teaching methodologies that are in place now focus mainly on programming skills instead of problem solving and design skills. To address these issues, we have explored a new pedagogical approach called studio-based learning. Adapted from architectural education, this instructional model emphasizes learning activities in which students (a) design computational solutions to problems that lend themselves to multiple solution strategies, and (b) present and justify their solutions to their instructors and peers for critical review and discussion.

In this research, three web-based systems (Peer Review System , Survey Data Collection System and Solution Upload System) were designed and built to support studio-based assignments and labs in two computing courses CS2 (COMP 2210) and CS3 (COMP3270). A web-based portal was also designed that enables the studio-based learning community to communicate and collaborate on a regular basis.

We performed data analyses to evaluate the effectiveness of studio-based instruction on students' learning and attitudes when compared to traditional instruction. Analyses of student performance data showed that students learned better in a full-fledged studio implementation when compared to a partial studio implementation of CS2. Analyses of student attitude and motivation survey data showed that students in the full-fledged studio implementation of CS2 had higher levels of sense of community, extrinsic motivation, self regulation and peer learning compared to the partial studio implementation.

However, we found that student learning was unchanged in CS3 between studio and traditional implementations. Also, though the critical thinking ability of students improved in the studio implementation, their extrinsic motivation, efficacy and sense of community decreased. Possible reasons for these findings and future work are discussed.

ACKNOWLEDGEMENTS

I express my deep sense of gratitude for the help and advice from my major mentor, Dr. N. Hari Narayanan without whose help this work wouldn't have seen the light of the day. My sincere thanks to my committee professors, Dr. Dean Hendrix and Dr. Margaret E. Ross for their advice and proof reading during my research and reporting. Thanks are also due to my friends and colleagues for extending their constant cooperation and moral support. I shall be failing in my duties if I forget the help and blessings of my parents who stood behind me in every walk of my stint at Auburn University. I also extend sincere thanks to the authorities of NSF for my Graduate Research Assistantship. Finally, I am thankful to the Almighty for giving me this worthwhile opportunity.

Style manual or journal used: APA LIKE

Computer software used: Microsoft Word 2007

TABLE OF CONTENTS

LIST OF TABLES	x
LIST OF FIGURES	xii
1. INTRODUCTION: PROBLEM SPECIFICATION AND SOLUTION OUTLINE	1
2. LITERATURE SURVEY	4
3. SBL DESIGN AND IMPLEMENTATION FOR CS2 AND CS3	8
3.1. Step-by-Step Implementation	9
3.2. Implementation of SBL in CS2.....	12
3.2. Implementation of SBL in CS3.....	14
4. DESIGNING SBL SOFTWARE.....	15
4.1. A portal for SBL community	15
4.2. SBL support systems for classroom.....	22
5. SBL EVALUATION IN CS2 AND CS3.....	33
5.1. CS2: Comparison of Two SBL Implementations	41
5.2. CS3: Comparison of Traditional and SBL Implementations.....	49
6. CONCLUSION AND FUTURE RESEARCH.....	57
REFERENCES	60
APPENDIX A. MSLQ SURVEY INSTRUMENT	63
APPENDIX B. CS2 PRE TEST	76

APPENDIX C. CS3 PRE TEST 87

LIST OF TABLES

Table 5.1.	Summary of pre-to-post test score comparison across two semesters of CS2.....	41
Table 5.2.	Summary of post-test score comparison across two semesters of CS2.....	42
Table 5.3.	Summary of pre-to-post survey score comparison for Fall 07 semester of CS2	43
Table 5.4.	Summary of pre-to-post survey score comparison for Fall 07 semester of CS2	44
Table 5.5.	Summary of post-to-post survey score comparison for Fall 07 semester and Spring 08 semester of CS2.....	45
Table 5.6.	Summary of pre-to-post test score comparison across two semesters of CS3.....	50
Table 5.7.	Summary of post-test score comparison across two semesters of CS3.....	50
Table 5.8.	Summary of pre-to-post survey score comparison for Fall 07 semester of CS3	51
Table 5.9.	Summary of pre-to-post survey score comparison for Spring 08 semester of CS3	52

Table 5.10. Summary of post-to-post survey score comparison for Fall 07 semester and Spring 08 semester of CS3	52
---	----

LIST OF FIGURES

Figure 3.1	Studio-based Learning approach	11
Figure 4.1.1	SBLP overall data flow	16
Figure 4.1.2	SBLP Discussion Board	19
Figure 4.1.3	SBLP Email Client	19
Figure 4.1.4	SBLP Member Message Board	20
Figure 4.1.5	SBLP Built-In Calendar	21
Figure 4.1.6	SBLP Event Reminder	22
Figure 4.2.1	Students Write Reviews Selection Page.....	27
Figure 4.2.2	Students Write Review Page	28
Figure 4.2.3	Students Read Reviews Selection Page.....	28
Figure 4.2.4	Students Read Review Page	29
Figure 4.2.5	Admin and Instructor Main Menu.....	29
Figure 4.2.6	Admin User Menu	30
Figure 4.2.7	Admin and Instructor Student's Management Menu	30
Figure 4.2.8	Admin and Instructor Group Management Menu	31
Figure 4.2.9	Admin and Instructor View Reviews Selection Page.....	31
Figure 4.2.10	Admin and Instructor View Reviews Page	32
Figure 4.2.11	New User Page	32

Figure 5.1	Comparison of pre-to-post test scores across the semesters of CS 2	42
Figure 5.2	Comparison of pre-to-post survey scores of intrinsic motivation across two semesters of CS 2.....	46
Figure 5.3	Comparison of pre-to-post survey scores of extrinsic motivation across two semesters of CS 2.....	46
Figure 5.4	Comparison of pre-to-post survey scores of efficacy across two semesters of CS 2.....	47
Figure 5.5	Comparison of pre-to-post survey scores of peer learning across two semesters of CS 2.....	47
Figure 5.6	Comparison of pre-to-post survey scores of critical thinking across two semesters of CS 2.....	48
Figure 5.7	Comparison of pre-to-post survey scores of self regulation across two semesters of CS 2.....	48
Figure 5.8	Comparison of pre-to-post survey scores of sense of community across two semesters of CS 2.....	49
Figure 5.9	Comparison of pre-to-post test scores across two semesters of CS 3.....	51
Figure 5.10	Comparison of pre-to-post survey scores of intrinsic motivation across two semesters of CS 3.....	58
Figure 5.11	Comparison of pre-to-post survey scores of extrinsic motivation across two semesters of CS 3.....	53
Figure 5.12	Comparison of pre-to-post survey scores of efficacy across two semesters of CS 3.....	54

Figure 5.13	Comparison of pre-to-post survey scores of peer learning across two semesters of CS 3.....	54
Figure 5.14	Comparison of pre-to-post survey scores of critical thinking across two semesters of CS 3.....	55
Figure 5.15	Comparison of pre-to-post survey scores of self regulation across two semesters of CS 3.....	55
Figure 5.16	Comparison of pre-to-post survey scores of sense of community across two semesters of CS 3.....	56

CHAPTER 1

INTRODUCTION: PROBLEM SPECIFICATION AND SOLUTION

OUTLINE

The field of computer science is known for its innovation and creativity. For one, it is experiencing a significant decline in incoming majors. According to the latest Taulbee survey, there is a 39 percent decline in the enrollment of computer science majors over the last two years [19] and at the same time, demand for college graduates with computing skills is on rise. According to the latest reports on changes in the field of computer education [19], the main reasons for failure are (a) computing is too often related to programming, and (b) introductory computing education is too closely tied to programming languages. The teaching methodologies which are in place now are focusing mainly on programming skills instead of teaching problem solving and design skills. To address these issues, we are proposing a new teaching methodology called studio-based learning.

Studio-based learning (SBL) is an instructional technique that emphasizes collaborative, design-oriented learning [1]. This pedagogy is not new; it dates back to old architectural schools where they have practiced this in the form of design studios [20] where (a) students created their own work spaces, (b) students worked in groups to solve problems, and (c) students presented their solutions to the class to obtain feedback from

their instructors and also from their peers. The design studio curriculum typically involves a series of problems, which may either be a sequence of more challenging design problems, or various components of a large project. A key aspect of design studio is design critique. Design critiques are review sessions where students present their work to the instructor and to the class for feedback.

Our new studio-based learning approach encompasses two key activities: 1) Students use various techniques such as algorithmic and data visualizations to describe their solutions. 2) Students present their personalized representations or visualizations to their instructors and peers for feedback and discussion. This approach also emphasizes the importance of presentations and discussions mediated by the students' representations. Thus, students participate in the course in ways usually reserved only for instructors and teaching assistants.

The implementation and evaluation of the studio-based learning methodology is a collaborative project of researchers from Auburn University, University of Hawaii and Washington State University [2]. At Auburn we have implemented the studio-based learning approach in two courses CS2 (Fundamentals of Computing II, COMP 2210) and CS3 (Introduction to Algorithms, COMP 3270). The research reported in this document was done as part of this multi-university project.

Key aspects of our studio-based learning approach are [3]:

- a) **Meaningful Problems:** Students are given meaningful problems for which they have to design and implement computational solutions in groups.
- b) **Multiple Strategies:** The problems given to students are amenable to multiple solution strategies. This means that the students have to consider alternate

solutions and their tradeoffs in terms of efficiency and software engineering considerations.

- c) Work in Groups: Students are divided into groups by the instructor. Groups have to work together on the problem to develop a solution or multiple solutions.
- d) Presentations: Students must then articulate their solutions and justifications to the entire class for peer review, feedback and discussion, in writing, orally or both.
- e) Peer Reviews: Their peers and instructor evaluate the solutions and provide comments and criticisms, again in writing, orally or both.
- f) Response: Students are given the opportunity to respond to the feedback they received and modify their solutions appropriately.

Through these steps the students get experienced in (1) individually and collaborative solving algorithm and software design problems, (2) evaluating and selecting strategies based on correctness, efficiency and other design issues, (3) explaining their solutions to others in writing and through oral presentations, (4) critically analyzing each others' solutions in peer reviews, and (5) learning from their mistakes.

CHAPTER 2

LITERATURE SURVEY

Studio-based learning is an instructional technique that emphasizes collaborative, design-oriented learning [2] and construction of personally meaningful representations of computer concepts and processes through an approach in which students solve complex problems, present their solutions to the class, and participate in critical discussion.. The model, adapted from architectural schools, involves a “design studio,” a place where students set up their own workspaces, and create and present their designs. As students work on design tasks in this common space, they develop a “community of practice,” providing support and feedback to each other. The design studio curriculum involves a series of design problems, which may either be a sequence of progressively more challenging design problems, or various components of a large design project. A key aspect of the design studio is the design critique. Design critiques are review sessions in which students present their evolving solutions to the instructor and the class for feedback and discussion. The following paragraphs discuss various past adaptations of this idea by computing educators.

Docherty et al. [17] describe a degree program, its curriculum and rationale, offered in studio format at the University of Queensland. This degree combines core computer science content with courses based on the architectural studio involving teamwork, collaborative learning, interactive problem solving, presentations and peer

review. Their evaluation studies concluded that students in the Information Environments Degree program overwhelmingly endorsed the studio-based approach to teaching and learning, because it provided them an opportunity to contextualize the skills they learnt in other courses and to apply them to real-life design problems.

Carbone and Sheard [14] describe the redesign of a traditionally delivered IT degree into a studio based teaching model. They also report on first year students' reactions to the new learning space, the IT tools and the change in teaching philosophy and a new method of assessment. They used a survey to determine how the students used the new environment when compared to the traditional lecture-tutorial style approach. They concluded that in general most of the first year students enjoyed learning in the studio environment, but also reported some concerns regarding the self-managed learning experience in the studio model. They also highlighted four aspects that should be considered when constructing future learning environments: physical space, teaching approach, assessment method and IT facilities.

The impact of visual representations and student interaction on learning in an upper level computer science course was studied [4, 7, 9, 10, 11, 12]. In this course, students performed the activities of representation construction, sharing, evaluation and discussion. Statistical analyses of pre- and post-test data indicated that student learning improved as a result of these activities.

Hundhausen [15] describes ALVIS, which is an algorithmic visualization system. The purpose of ALVIS is to involve students in creating low fidelity algorithm visualizations. This system enables students to (a) quickly construct rough, unpolished visualizations in much the same way they would with simple art supplies, and (b)

interactively present those visualizations to an audience. This system uses a language called SALSA that enables the layout and logic of algorithmic visualization to be specified in terms of its spatiality. ALVIS was designed using the following design principles:

- Users must be able to create, systematically layout, and animate simple objects containing sketched graphics.
- Users must be able to construct storyboard objects by cutting and sketching; they must be able to position objects by direct placement.
- Users must be able to create storyboards using spatial relations, not Cartesian coordinates.
- The system must support an execution model based on spatial, rather than algorithmic logic.
- The system must enable users to present their storyboards interactively. This entails an ability to execute storyboards in both directions; to rewind and fast forward storyboards to points of interest; and to point to, mark-up and modify storyboards as they are being presented.

Hundhausen et al. [5, 6, 13, 16] studied the effects of the studio-based approach on learning, in which students created and discussed visualizations of computer algorithms in lower level undergraduate computer science courses using ALVIS and SALSA. They concluded that the construction and discussion of visual representations by students improved their learning, and increased their sense of community, motivation and attention.

Hübscher and Narayanan [8] described CAROUSEL (Collaborative Algorithm Representations of Undergraduates for Self-Enhanced Learning). CAROUSEL is a web based system used by students to create expository representations of algorithms, share their representations and evaluate each other's representations. The web interface created allows instructors to set up schedules, post activity information and provide feedback to students about their performance. Once students submit their representations, the system allows them to evaluate others' representations based on six characteristics on a five-point Likert scale. These characteristics are usefulness, understandability, salience, familiarity, pleasure and originality. The authors also conducted three empirical studies to find out the effect of CAROUSEL on the algorithm learning abilities of students. They concluded that activities of representation construction, sharing, evaluation and discussion helped students better understand algorithms.

CHAPTER 3

SBL DESIGN AND IMPLEMENTATION FOR CS2 AND CS3

We have implemented the studio-based learning approach in two computer science courses at Auburn University. This first course in this series of two is Fundamentals of Computing II (CS2) (COMP 2210). Students learn about data structures like arrays, trees etc., and algorithms that solve problems using data structures. Students meet twice a week for 75 minutes for in-class lectures and twice a week for 75 minutes in a laboratory where they work on their individual programming assignments with the help of a teaching assistant. The second course is Introduction to Algorithms (CS3) (COMP 3270), in which students learn about algorithms for standard computational problems and techniques for analyzing their efficiency; designing efficient algorithms and experimentally evaluating their performance. Students meet twice a week for 75 minutes for in-class lectures and this course does not have any laboratory component, so the students have to work on the assignment problems outside the class. In Fall 07 semester we offered the CS2 course in a pilot studio-based learning format and CS3 in the traditional format. In Spring 08, CS2 was offered in a full studio format and CS3 was offered in a pilot studio format.

We implemented and evaluated the studio-based learning approach to address the following research questions:

- Do students learn better in studio-based instruction than in traditional instruction?
- Are they more engaged, invested and motivated due to the new pedagogical model?

3.1 Step-by-Step Implementation

1) **Informed Consent:** In the beginning of a semester, students are briefed about the research project and a consent form is given to them to sign if they are interested in participating in the project. Five percent extra credit is offered to students who participate in the research. The participation of students is voluntary.

2) **Pre-Attitude Survey:** In the beginning of the semester, the students take an attitude and motivation survey called the MSLQ (Motivated Strategy Learning Questionnaire) [18], used to evaluate their level of motivation, critical thinking, efficacy, peer learning and sense of community.

3) **Pre-Test:** All the students who enroll in the course are required to take a pre-test with thirty questions, out of which 20 are multiple choice questions and 10 are descriptive questions. The pre-test is used to measure the prior knowledge of students coming into the course.

4) **Studio Labs or Assignments:** Teaching the course in the studio format affects how the course labs or assignments are done. Figure 1 explains this process.

In the traditional approach, students used to work individually to solve lab or assignment problems. In the studio approach, students work in groups (. The instructor divides the students into groups randomly or based on their class performance.

Students are given complex and meaningful problems in their labs or assignments, for which they have to design and implement computational solutions. Then they have to come up with multiple strategies to solve the problem and choose one based on tradeoffs in terms of efficiency and software engineering considerations.

Then they are asked to present their strategies and solution designs to the entire class for peer review, feedback and discussion in writing, orally or both. After the presentations, students have the opportunity to revise their strategy, write the code, and submit both for reviews by others. Then each student has to individually review the work of other students assigned to him by the instructor. This peer review process is anonymous. Finally, each and every student gets a chance to respond to reviews received from other students.

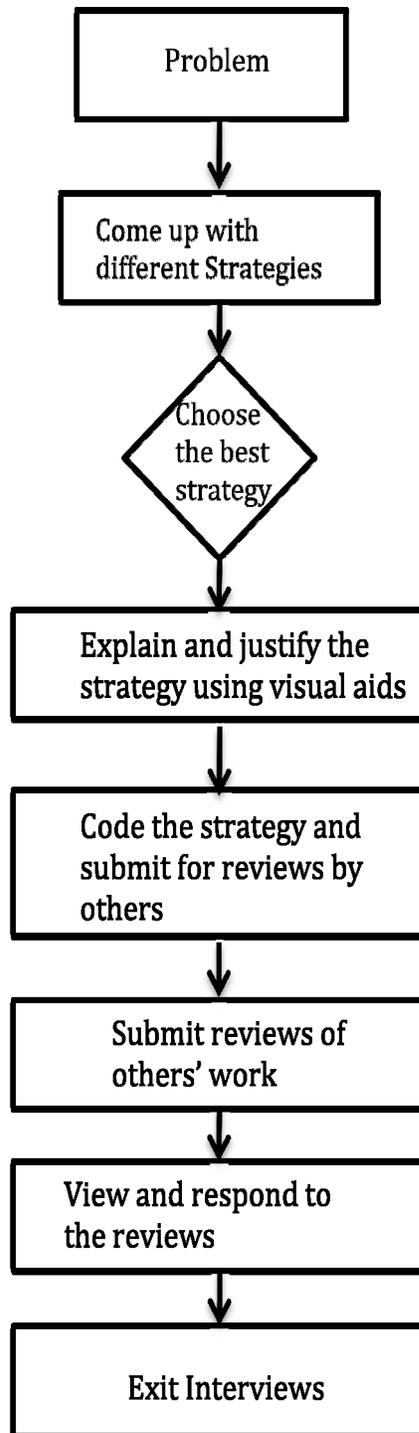


Figure 3.1 Studio-based Learning approach

5) **Post-Attitude Survey:** At the end of the semester, the students complete a survey that is identical to the pre-attitude survey.

6) **Post-Test:** The students take a post-test at the end of the semester as part part of their final exam. This test is identical to the pre-test.

7) **Exit Interviews:** Exit interviews are conducted to obtain feedback from the students on the studio-based methodology. Five students are selected per course: two from the top one percent, one from the middle level of performance and two from the bottom one percent. The interviews are recorded.

This implementation was followed in both courses with some variation.

3.2 Implementation of SBL in CS2

Traditionally, the instructor would assign programming problems in this class; students would work on these outside the class and in labs, and submit their solutions, which were graded for correctness and efficiency by teaching assistants. In Fall 07 we first implemented the SBL model in this course, i.e., five out of six assignments were designed to have the features of SBL described above. The assignments remained individual assignments. Each included a problem (e.g., a problem called "infinite monkeys") that could be solved with multiple strategies. Students were asked to first think about various strategies, choose one, and explain the strategy they chose and justify it in writing, using text and pictures. Their submissions of strategy explanations, visualizations and justifications were made anonymous and provided to the entire class for viewing on the web. In addition, each student was assigned to some other students' submissions to review. Students submitted these peer reviews in writing through the web.

Following this, the students implemented their strategy in Java and submitted their code for grading. Finally, each student was given the option of orally responding to criticisms of his/her strategy in a studio session conducted in the course lab. We videotaped and analyzed these sessions.

This studio implementation revealed several problems, such as: (1) students had difficulty in understating some of the questions they were required to answer as part of the peer reviews, (2) accurately and fully recording student responses in the labs proved difficult, and (3) students were confused as to exactly what was expected in their responses to critiques.

Based on our observations and student feedback we received in Fall 07, we made many design changes to the SBL approach for Spring 08 semester. We classified the Fall 07 implementation as pilot SBL and the revised Spring 08 implementation as complete SBL. The major design changes we made are: (1) the lab assignments were turned into group assignments, (2) students presented their intermediate solutions to the class so that they could use the class feedback to improve their solutions prior to submission and these presentations were videotaped, and (3) the response to critiques step was removed.

The span of each lab assignment was over 3 weeks, and students worked in groups of two for all assignments in the Spring 08 semester. The group composition was changed for every home work by the instructor. In the first week of an assignment, students had to come up with multiple strategies for a given problem and choose one. Students were then asked to present their strategy to the whole class, which was videotaped, and then they had to implement their strategies in java and submit for grading. Each student was then assigned up to four submissions to be peer reviewed.

3.3 Implementation of SBL in CS3

In spring 08, we implemented the studio-based approach in CS3, in which 3 out of 6 assignments were designed to have SBL features. The first three assignments were in the traditional format (individual problem solving assignments) and the last three were in the SBL format. The span of each SBL home work was over 3 weeks, and students worked in groups of five. The instructor formed the groups based on the performance of students, and changed the group composition for every home work. Rest of the implementation of the SBL model in CS3 assignments was exactly the same as the implementation in CS2 labs as described above. The students presented their strategies to the whole class during lecture hours, unlike in CS2 where the students used their lab time.

This studio implementation revealed several problems, such as: (1) students complained about lack of sufficient time to work on the problems, (2) students complained about the large group size and unequal work distribution due to some group members not being responsible, and (3) students had issues with the implementation of SBL, as it was implemented only from the mid-semester. Based on our observations and student feedback, we classified the Spring 08 CS3 as pilot SBL and made many design changes for the next studio implementation, which was done in Fall 08, and is outside the scope of this research.

This chapter described the design and implementation of the studio-based learning approach in CS2 and CS3 during the 2007-08 academic year. The next chapter will describe the design and implementation of SBL software used to support studio-based assignments and labs in the two computing courses CS2 and CS3.

CHAPTER 4

DESIGNING SBL SOFTWARE

4.1 A Portal for the SBL Community

I designed and implemented a web-based portal that enables the studio-based learning community to communicate and collaborate on a regular basis. It establishes a web-based community system that (a) facilitates on-line communication among core researchers, computing educators who are community members and the general public, (b) provides a repository of curricular materials and a collaborative workspace in the form of a wiki that all community members can access and contribute to, and (c) presents an information dissemination website on SBL to the public.

Architecture

The Studio-Based Learning Portal (SBLP) is a web-based collaborative system. Users can access SBLP, if they have an internet connection and a web browser. The system that hosts SBLP is a Linux PC with Apache as the web server and Mysql as the database server. Macromedia Dreamweaver MX 2004 was used as the software development environment and PHP was the programming language. SBLP has two kinds of users, public users and registered member users. The public users can access the SBLP without any login information, but they are restricted to only the public area of the portal, whereas the member users can access the private areas by logging in into the portal with their username and password. Public users can register at the SBLP web page to become

a member of the Studio-Based Learning Community. Their login information is recorded into the database. Once they are logged in, the database records all user interactions with timestamps; it even captures the IP address, hostname of the visitor and stores this information in the database. Figure 4.1.1 below explains the data flows in SBLP.

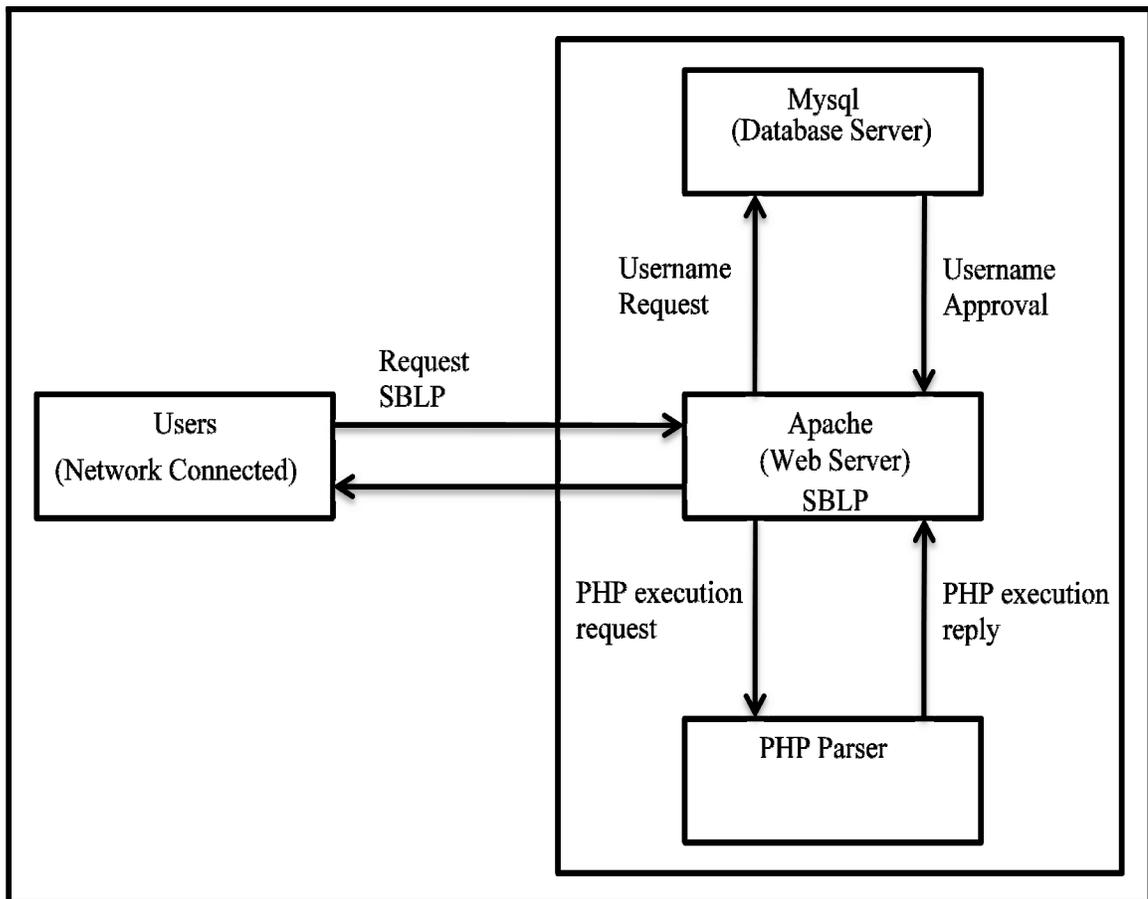


Figure 4.1.1 SBLP overall data flow

Member Levels

The Studio-Based Learning Web Portal (SBLP) has two kinds of users.

1. Public User

Members of the public who are interested in Studio-Based Learning and who do not belong to the SBL community come under this category. The information and features accessible to public users are listed below.

- a) Pages describing the studio-based approach.
- b) A discussion board to post comments or ask questions.
- c) Links to support systems that have been built and used in various courses at various universities implementing studio-based learning.
- d) Links to web sites of courses at various universities and colleges.
- e) A community directory of those members who have chosen to make their information public, and to whom emails can be sent directly from the portal.
- f) A link for interested members of the public to join the community.
- g) A way to send emails to the registered members and portal administrator.

2. Registered user

Those who join the studio-based learning community become registered members of SBLP. The portal provides the following facilities to registered members through login access.

- a) Every community member gets a home page in the portal and a member can upload information, announcements and files that will automatically appear on every member home page so that this information will be seen whenever anyone logs in.
- b) Members can upload resources specific to a computing course, and search for specific resources contributed by other members.
- c) A wiki to be used for collaborating on material creation and sharing amongst community members.

- d) An internal communication mechanism through which community members can engage in email discussion directly from the portal without having to use an external email client. They can send messages to specific members, the CPATH team of investigators, or the entire SBL community, and can choose whether to have their messages appear on the public discussion board.
- e) An activity planning area with a calendar where conference related activities, community workshops, and teleconferences will be planned and announced.
- f) A reminder feature where members can set up their own reminders of scheduled events.

Key Features

The SBLP has various built in features to be used by registered users and a few features available to public users. The key features of SBLP are discussed below.

1. Resource Sharing:

This feature helps members of the Studio-based Learning Community share resources, such as curricular materials and test instruments, among themselves.

2. Discussion Board:

The online discussion board helps the public users interested in Studio-Based learning to post a comment or ask a question to the core investigators or the larger community, and the question and answers provided by multiple respondents will appear on the discussion board.

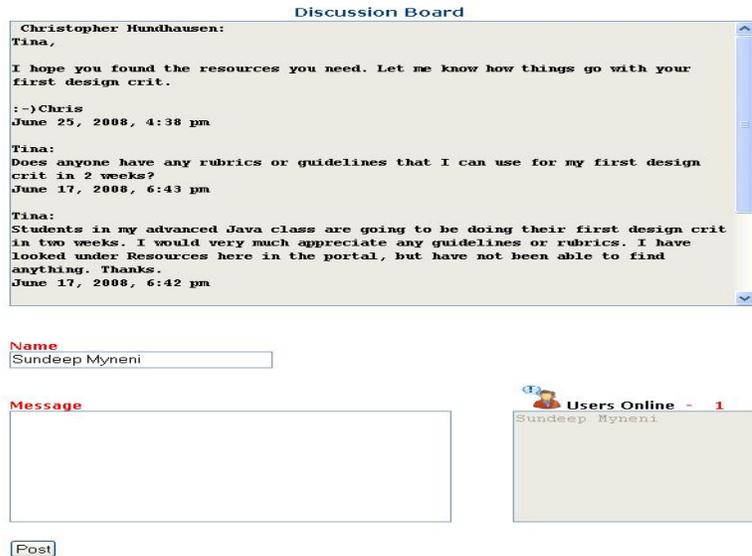


Figure 4.1.2 SBLP Discussion Board

3. Online Communication:

The SBLP enables online communication in the form of emails. This feature is available only to registered members of the SBL community.

Write Message	
From	<input type="text" value="sundeep.myneni@gmail.com"/>
To	<input type="text" value="dddd@test.com"/>
Subject	<input type="text" value="RE: dddd"/>
Text	<div style="border: 1px solid black; padding: 5px;"> <p>----- dddd</p> </div>
<input type="checkbox"/> Display in the Discussion Board	
<input type="checkbox"/> Send email to Community	
<input type="checkbox"/> Send email to Core Group	
<input type="checkbox"/> Send email to individual member	
<input type="checkbox"/> Anu <input type="checkbox"/> Crosby <input type="checkbox"/> Hendrix <input type="checkbox"/> Hundhausen <input type="checkbox"/> Myneni <input type="checkbox"/> Narayanan <input type="checkbox"/> Ross <input type="checkbox"/> Trevisan <input type="checkbox"/> Vick	
<input type="button" value="Send"/>	

Figure 4.1.3 SBLP Email Client

4. Resource Repository for Group Collaboration:

A wiki embedded in the SBLP acts as a resource repository, where community members can find, upload and edit documents and other kinds of files. This repository is available only to registered community members. The resource sharing feature mentioned before allows easy exchange of files. The wiki, on the other hand, facilitates collaborative construction and editing of files as well as sharing.

5. Message Board:

A message board is located on the home pages of all registered members, where we can display important information that they will see each time they log in.

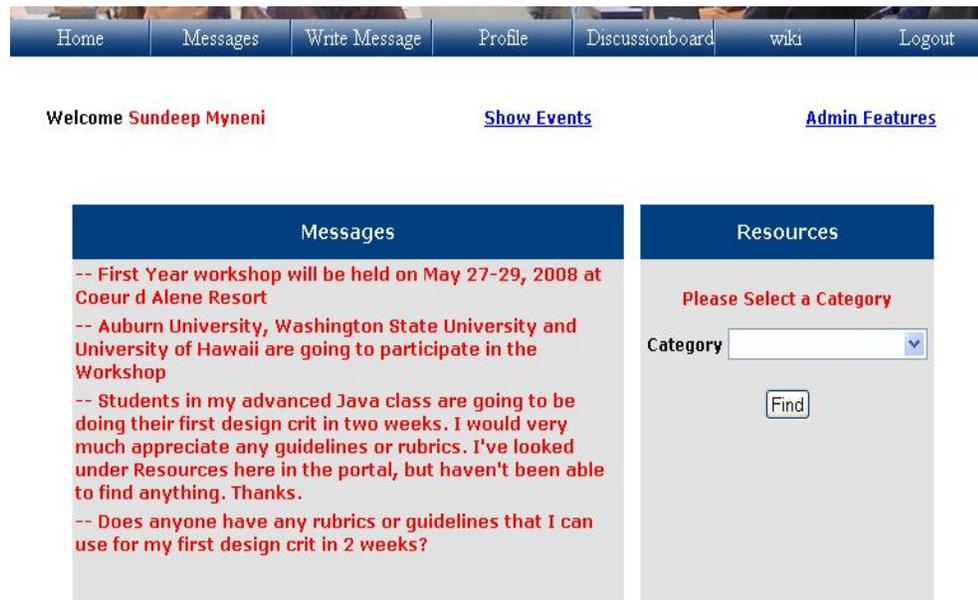


Figure 4.1.4 SBLP Member Message Board

6. Event Calendar

The portal has a built in calendar to schedule events. Only registered users can schedule the events, which are visible to all the registered members of the portal.

July 2008

[Set Event Reminders](#) [Time Zones](#)

<< July 2008 >>

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Figure 4.1.5 SBLP Built-In Calendar

7. Event Reminders

The portal has an event reminder feature where one can setup one's own private reminders or common reminders that one wants every registered member of the community to receive. The system sends these reminders, to one individual (in case of private reminder) or to the entire community (a group reminder) via email one day prior to the scheduled event.



Figure 4.1.6 SBLP Event Reminder

4.2 SBL support systems for classroom

Studio-Based Learning Support Systems (SBLSS) used for this research consist of a Peer Review System (PRS) and a Survey Data Collection System (SDCS). The peer review system is built along the lines of an earlier system called CAROUSEL. CAROUSEL is a Computer Supported Collaborative Learning (CSCL) system, which allows students to display their designs and evaluate each others' designs [8]. It also has features for instructor of a course to help manage course activities.

Peer Review System (PRS)

The Peer Review System is designed to support students engaging in the activities of constructive and collaborative learning. This system allows students to upload their work into the system, share it with others, and to write reviews of others' submissions. PRS is a web based system implemented in PHP and the server runs on a Linux PC with Apache as the web server and Mysql as the database driver. The architecture of the system is similar to the one described in section 4.1 (Fig: 4.1.1). The web-based interface

created allows students to submit their work, and to provide feedback and comments on other students' work. This interface also helps the system administrator to create student groups and peer review questions.

User Profiles

The Peer Review System (PRS) has three kinds of users: students, instructors and administrators. All users of the system have editable system profiles. The system profile of a student user consists of the following components: a posting code that is used to hide the student's identity, email, name and password. The system requires students to login, so that they can write reviews and comments on another's work. A student can see only the work of the student assigned to him or her for reviewing; where as in CAROUSEL a student could see the work of all other students. The instructor is the one responsible for creating peer review assignments (which student will review which other student's or group's work) and peer review questions that students will use to assess the work of others. Once he logs in as an instructor, he has the rights to create the review assignments and questions. He can also see the reviews written by all students, but he is restricted from deleting anything from the system. The administrator has full access to the system, and he can control everything using a web-based administrator interface developed in PHP.

The system provides tools for an instructor to create review assignments (Figure 4.2.8), write review questions, send messages to individual students, see the reviews of all students (Figure 4.2.9), and inspect the review of an individual student. In this way, PRS gives the instructor the ability to provide feedback to students about their reviews.

The administrator has additional rights such as the delete feature (Figure 4.2.6) and access to a log file (Figure 4.2.5) that stores every navigation step of a logged in student.

Managing Reviews

The student first creates a login for himself or herself using the Posing Code provided by the instructor (Figure 4.2.11). Posting Code is a unique id assigned to each and every student by the instructor. To keep their work anonymous, students include only their posting codes, not names, in their submissions. For every assignment, the instructor will create student groups. All assignments are thus team-based, whereas the peer review is done by individual students. The instructor also creates up to 10 review questions that a student has to answer in order to complete his or her review of the work of another student or group. Some of these questions may remain the same for all assignments throughout the semester while others may be assignment-specific. These review questions encourage the student to evaluate the explanations, strategies, strategy correctness, strategy efficiency and source code when reviewing the work of others. Each student in a class is expected to review the work of at least another student/group as part of each assignment.

The two courses which make use of this Peer Review System at Auburn University are Fundamentals of Computing II (CS2) and Introduction to Algorithms (CS3). Once the students finish their homework, they upload their files into the system, which are not accessible to other students until the homework deadline has passed. Meanwhile the instructor or the administrator creates the groups and reviews questions for that particular homework and makes the files accessible to students after the deadline.

An email will be sent to the students as soon as the system is ready for peer reviews. Then students login into the system and they go to the "write reviews" area (Figure 4.2.1), where they select the appropriate homework number and go to a page that displays the correct set of review questions. There is a drop down list available on top of the page where they can locate the posting codes of other students whom they have to review, thus keeping the review process anonymous for both the reviewer and the reviewed (Figure 4.2.2). Students then go to the files section and read the homework submissions of the students assigned to them for reviews, and they submit the reviews by filling in text boxes associated with the peer review questions.

They have a "history" tab (Figure 4.2.5) in the menu where they can track the progress of their reviews and also a "read reviews" tab (Figure 4.2.4) where they can read the reviews given to them by other students.

Survey Data Collection System (SDCS)

Survey Data Collection System (SDCS) is a web-based online survey that is used to collect data related to student attitudes and perceptions. The architecture of the system is similar to the one described in section 4.1 (Figure 4.1.1) and it was developed in PHP with Mysql as database. The entire survey consists of 129 questions divided into three parts, each of which collects a specific type of data. The first part is a demographics section with 11 questions - name, gender, age, class level, ethnic background, major, number of courses taken in engineering, number of hours spent in a week to study engineering subjects, number of hours spent in a week for a particular engineering course, level of confidence and usefulness of the course. The second part of the survey

contains 81 questions dealing with motivational aspects, and the third part contains 37 questions related to sense of community.

The survey instrument is based on the Motivated Strategies for Learning Questionnaire (MSLQ, National Center for Research to Improve Postsecondary Teaching and Learning) and the Sense of Community Questionnaire (Washington State University). The MSLQ was designed to assess college students' motivational orientations and their use of different learning strategies for college courses. There are mainly two sections to MSLQ, a motivation section and a learning strategies section. The motivation section consists of 31 questions that assess students' goals and value beliefs for a course, their beliefs about their skill to succeed in a course, and their anxiety about tests in course. The learning strategy section contains 31 questions regarding students' use of different cognitive and metacognitive strategies. The learning strategy section also includes 19 questions concerning student management of different resources.

Students rate themselves on a seven point Likert scale from "not at all true of me" to "very true of me". There are fifteen different scales in the MSLQ survey and scales are constructed by taking the mean of the questions that make up the scale. For example, the Self-Efficacy scale is based on eight questions (5, 6, 12, 15, 20, 21, 29, 31 Part II MSLQ Questionnaire). An individual score for Self-Efficacy would be computed by adding the eight questions and taking the average. The other fourteen scales contained in MSLQ are: intrinsic motivation, extrinsic motivation, critical thinking, peer learning, sense of community, self regulation, task value, learning benefits, test anxiety, rehearsal, elaboration, organization, time and study environment and effort regulation.

Some of the questions marked as “reversed” are reverse coded items and must be reflected before scale construction. These negatively framed questions and ratings have to be reversed before an individual’s score can be computed. If a question is to be reversed, a person who has circled 1 for that question with a seven-point Likert scale receives a score of 7 and so on.

This chapter described the two web-based support systems that were designed to support studio-based assignments and labs in the two computing courses CS2 and CS3. The next chapter will describe the evaluation of studio-based learning in CS2 and CS3.

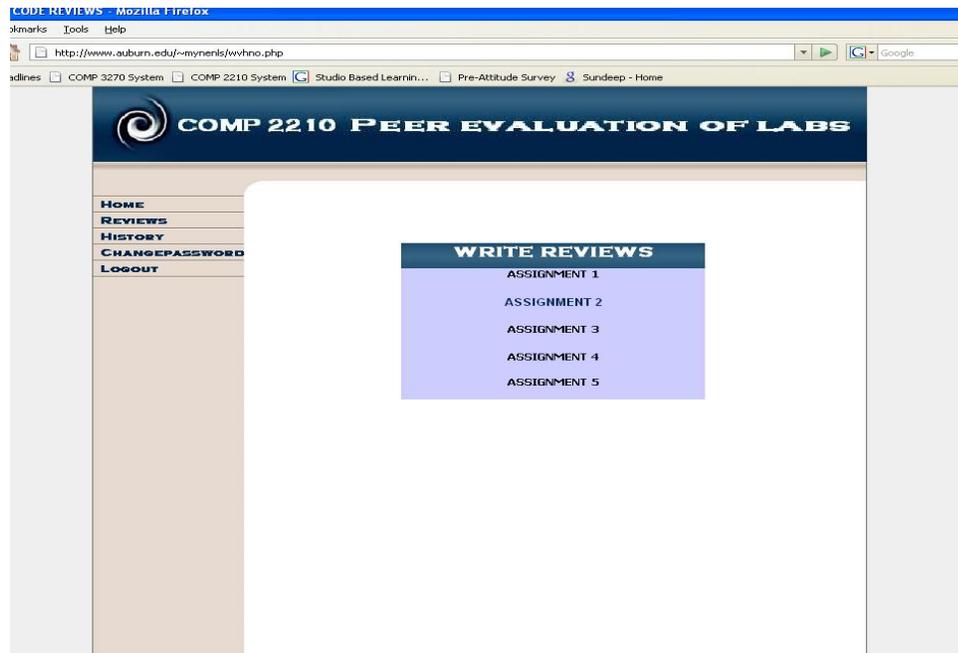


Figure 4.2.1 Student Write Reviews Selection Page

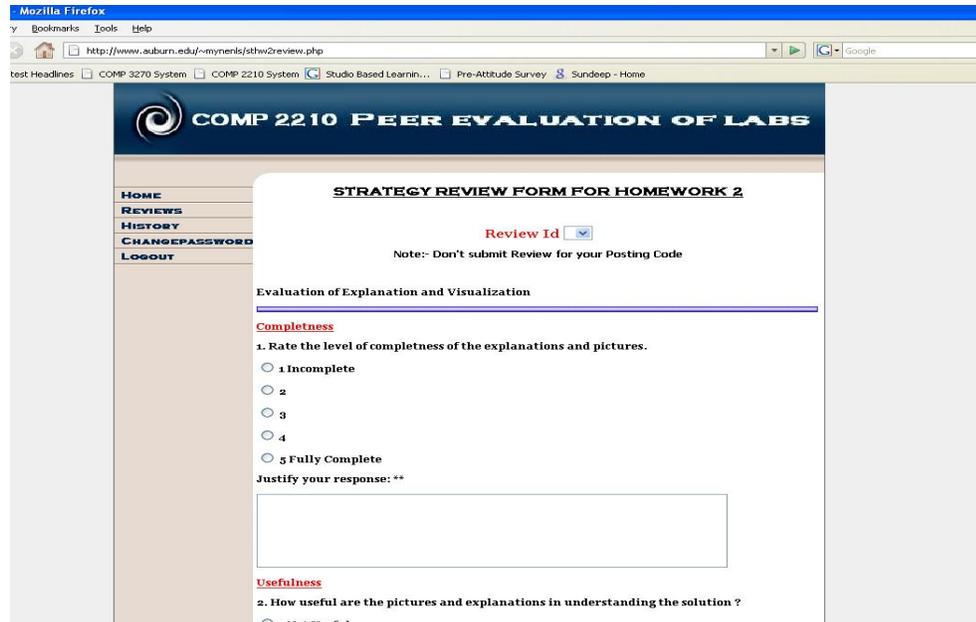


Figure 4.2.2 Students Write Review Page

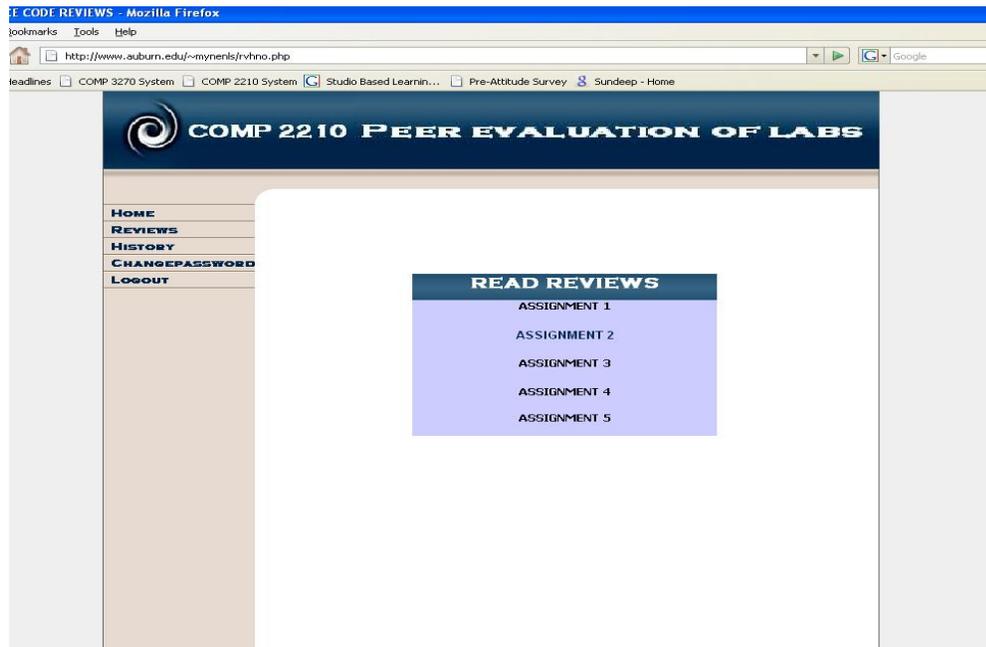


Figure 4.2.3 Students Read Reviews Selection Page

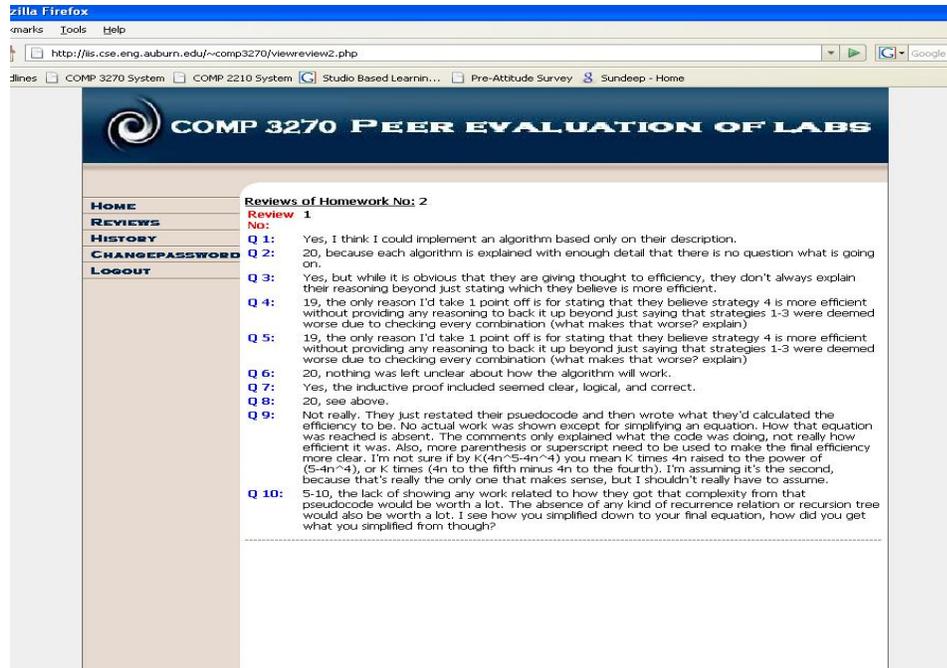


Figure 4.2.4 Students Read Review Page

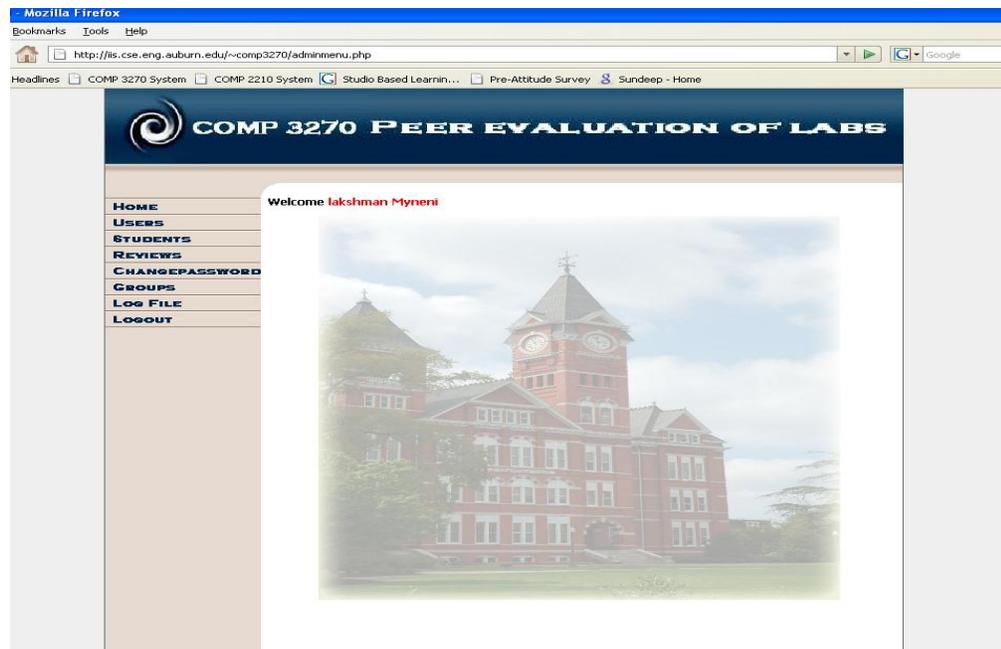


Figure 4.2.5 Admin and Instructor Main Menu

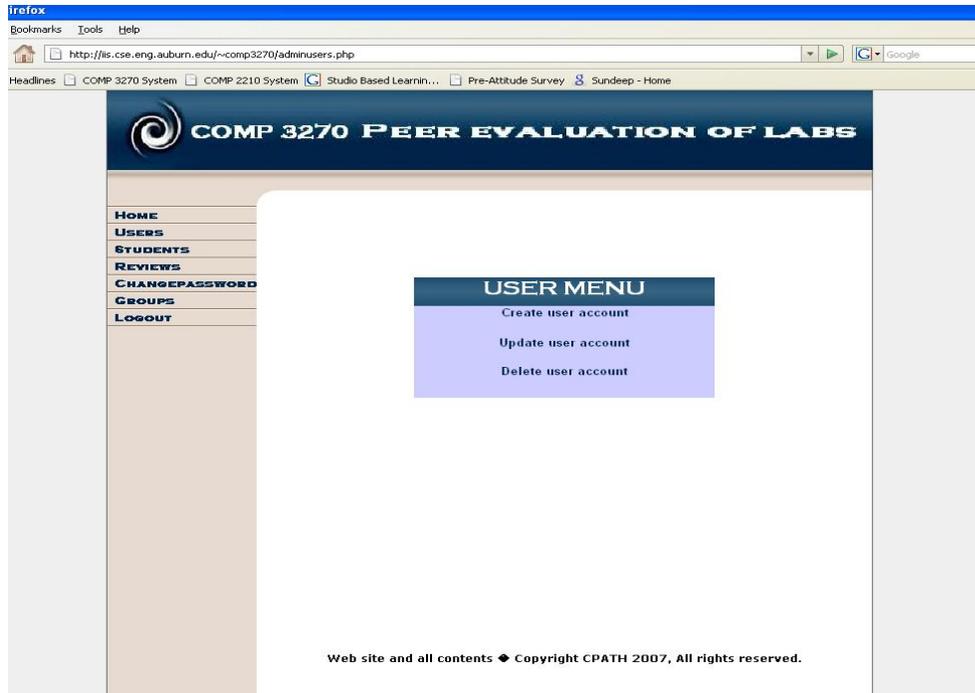


Figure 4.2.6 Admin User Menu

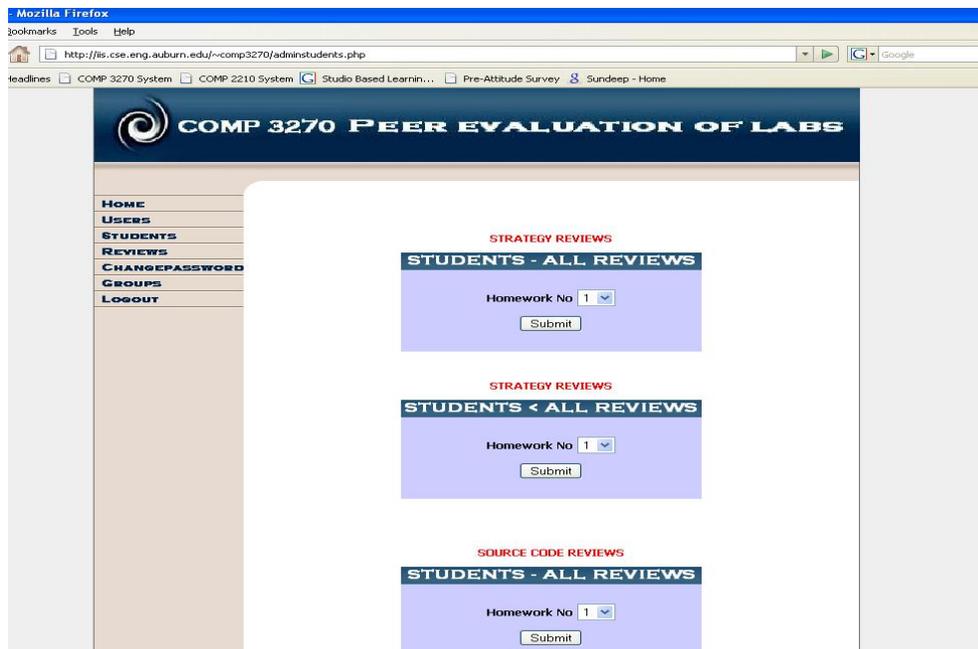


Figure 4.2.7 Admin and Instructor Student's Management Menu

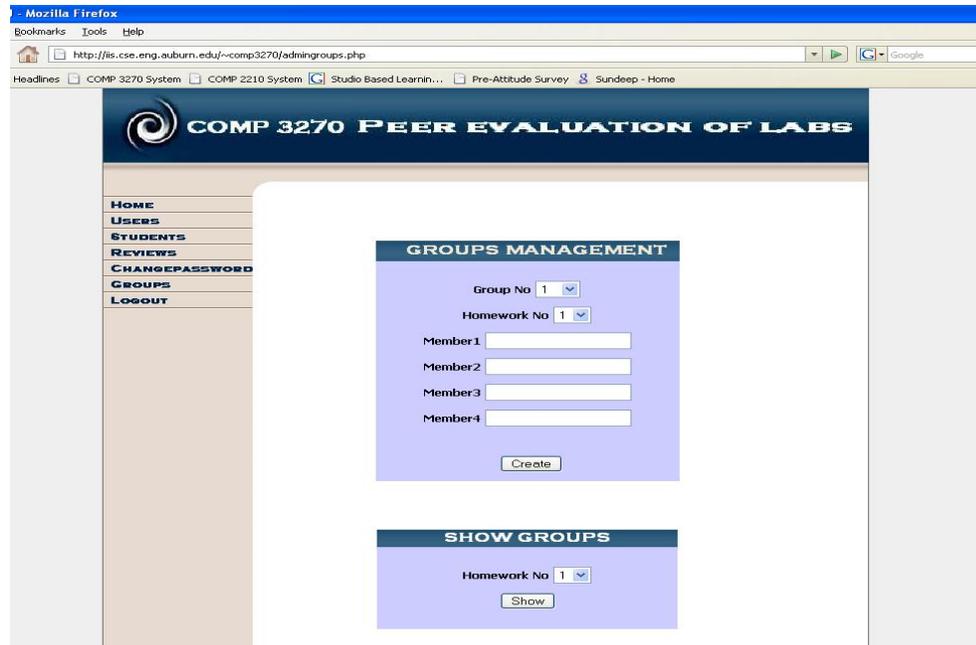


Figure 4.2.8 Admin and Instructor Group Management Menu

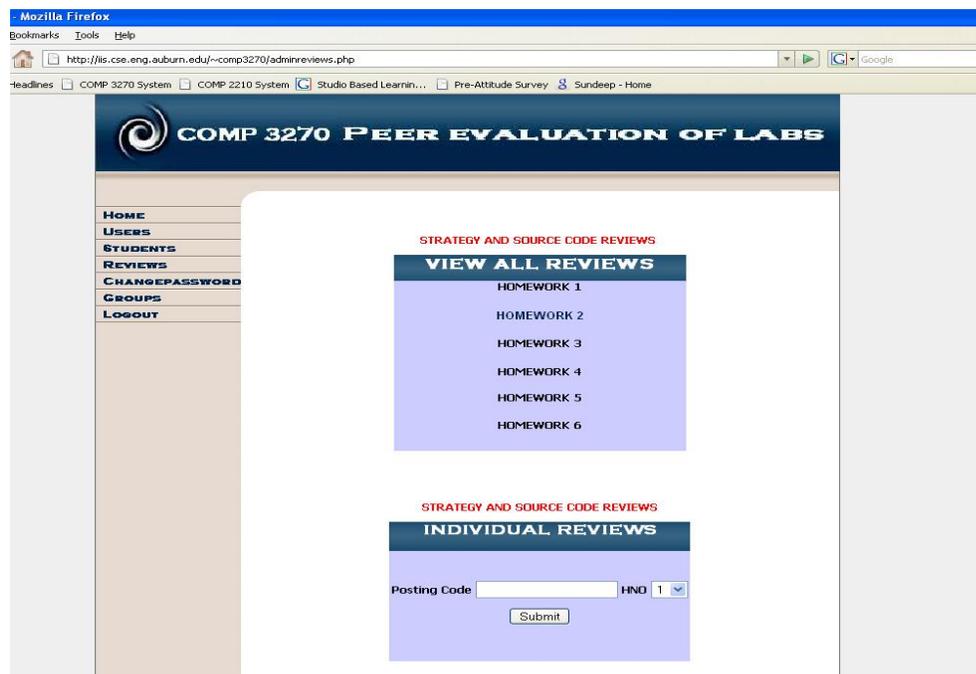


Figure 4.2.9 Admin and Instructor View Reviews Selection Page

COMP 2210 PEER EVALUATION OF LABS

HOME
USEDS
STUDENTS
REVIEWS
CHANGE PASSWORD
GROUPS
LOGOUT

Reviews of Homework No: 2
Review No: 1
Student ID: 2F085F208
Time: September 25, 2008, 9:58 pm
Review ID: 2F085M203
Completeness: 1
Comments: There wasn't a design powerpoint in the zip file.
Usefulness: 1
Comments: There wasn't a design powerpoint in the zip file.
Strategy: 1
Correctness: 1
Comments: There wasn't a design powerpoint in the zip file.
Efficiency: 1
Comments: There wasn't a design powerpoint in the zip file.
Redability: 3
Comments: There are not enough comments but there is enough indentation to be able to follow the code. Also, the names of the variables allow for easy readability in order to get a sense of what that variable's function is.
Understandability: 4
Comments: There is a little bit of confusion at the beginning of the main method when the text is being read in from the file and placed into 5 character groups but ultimately it is easy to understand what the code is basically doing and easy to follow.
Correctness: 1
Comments: When I ran the program it threw an `ArrayIndexOutOfBoundsException`. Also, to my knowledge, the first encryption of the word JOKER was incorrect.
Question 8: There are numerous while loops of time complexity $O(1)$ within the "JokersWild" method. There are no for loops and I believe the big-oh time complexity (worst-case) is $O(1)$ because there are no "n" variables, or a possible of "n" numbers of iterations. I maybe misunderstanding the big-oh but I believe its $O(1)$.
Question 9: There was no design attached to this zip file but I believe their strategy was to use arrays. I believe that it was a very efficient way of storing the deck of cards and manipulating it. The programmer did not consider all possibilities of the deck of cards which I believe led to the malfunction at runtime. I believe their method of splitting the characters into 5 groups and putting them into arrays was a little complicated and could be more efficient.
Question 10: I'm not sure how to improve the time complexity of this source code.

Review No: 2
Student ID: 2F08FM403
Time: September 25, 2008, 10:10 pm

Figure 4.2.10 Admin and Instructor View reviews Page

COMP 2210 PEER EVALUATION OF LABS

HOME
LOGIN
NEW USER
INSTRUCTIONS
FORGET PASSWORD
CONTACT US

NEW USER REGISTRATION

Email
 First Name
 Last Name
 Posting Code
 Password
 Confirm Password

Web site and all contents © Copyright CPATH 2007, All rights reserved.

Figure 4.2.11 New User page

CHAPTER 5

SBL EVALUATION IN CS2 AND CS3

Methodology

This chapter provides a description of the research methodology used in the evaluation of studio-based learning. The purpose of this study was to examine the effectiveness of studio-based instruction as compared to traditional instruction. The evaluation presented in this chapter addresses the following research questions:

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

RQ2. Are students more engaged, invested and motivated in classrooms employing the new pedagogical model?

Materials

A web-based online system called Survey Data Collection System (SDC) was used to collect data related to student attitudes (described in chapter 2). Students took the survey at the beginning of each semester (Pre-Survey) and again at the end of the semester (Post-survey). The online survey was developed from the MLSQ survey [18] and contains 129 questions. We have chosen six scales from the MSLQ and the sense of community scale to address the second research question for the purpose of this study.

The six scales are:

(1) Intrinsic Motivation

Items

- (1) In a class like this, I prefer course material that really challenges me so I can learn new things.
- (16) In a class like this, I prefer course material that arouses my curiosity, even if it is difficult to learn.
- (22) The most satisfying thing for me in this course is trying to understand the content as thoroughly as possible.
- (24) When I have the opportunity in this case, I choose course assignments that I can learn from even if they don't guarantee a good grade.

(2) Extrinsic Motivation

Items

- (7) Getting a good grade in this class is the most satisfying thing for me right now.
- (11) The most important thing for me right now is improving my overall grade point average, so my main concern in this class is getting a good grade.
- (13) If I can, I want to get better grades in this class than most of the other students.
- (30) I want to do well in this class because it is important to show my ability to my family, friends, employer, or others.

(3) Self-Efficacy

Items

- (5) I believe I will receive an excellent grade in this class.
- (6) I'm certain I can understand the most difficult material presented in the readings for this course.
- (12) I'm confident I can understand basic concepts taught in this course.
- (15) I'm confident I can understand the most complex material presented by the instructor in this course.
- (20) I'm confident I can do an excellent job on the assignments and tests in this course.
- (21) I expect to do well in this class.
- (29) I'm certain I can master the skills being taught in this class.
- (31) Considering the difficulty of this course, the teacher, and my skill, I think I will do well in this class.

(4) Peer Learning

Items

- (34) When studying for this course, I often try to explain the material to a classmate or friend.
- (45) I try to work with other students from this class to complete the course assignments.
- (50) When studying for this course, I oftenest aside time to discuss the course material with a group of students from the class.

(5) Critical Thinking

Items

- (38) I often find myself questioning things I hear or read in this course to decide if I find them convincing.
- (47) When a theory, interpretation, or conclusion is presented in class or in the readings, I try to decide if there is good supporting evidence.
- (51) I treat the course material as a starting point and try to develop my own ideas about it.
- (66) I try to play around with ideas of my own related to what I am learning in this course.
- (71) Whenever I read or hear an assertion or conclusion in this class, I think about the possible alternatives.

(6) Self Regulation

Items

- (33) During class time I often miss important points because I'm thinking of other things.
- (36) When reading for this course, I make up questions to help focus my reading.
- (41) When I become confused about something I'm reading for this class, I go back and try to figure it out.
- (44) If course materials are difficult to understand, I change the way I read the material.

- (54) Before I study new course material thoroughly, I often skim it to see how it is organized.
- (55) I ask myself questions to make sure I understand the material I have been studying in the class.
- (56) I try to change the way I study in order to fit the course.
- (57) I often find that I have been reading for class but don't know what it was all about.
- (61) I try to think through a topic and decide what I am supposed to learn from it rather than just reading it over when studying.
- (76) When studying for this course I try to determine which concepts I don't understand well.
- (78) When I study for this class, I set goals for myself in order to direct my activities in each study period.
- (79) If I get confused taking notes in class, I make sure I sort it out afterwards.

A second survey was used to measure students' self reported sense of community.

(7) Sense of Community

Items

- (1) Students in this class know each other.
- (2) I often find that I have been reading for class but don't know what it was all about.
- (3) Students in this class socialize with each other before and after class

- (4) When studying for this course I try to determine which concepts I don't understand well.
- (5) When I study for this class, I set goals for myself in order to direct my activities in each study period.
- (6) Students in this class help one another, e.g., taking notes if one has to be absent, etc.
- (7) Students in this class feel connected to each other.
- (8) Students in this class feel comfortable borrowing and lending things from each other (books, notes, etc.)
- (9) Students in this class think of themselves as a community.
- (10) The students in this class help each other out when someone has a problem.
- (11) Students in this class do things together outside of class.
- (12) Students in this class support one another.
- (13) Students in this class can make it a better course.
- (14) Students in this class trust each other.
- (15) Students in this class would comfort each other when someone does poorly.
- (16) Students in this class hang out together on campus.
- (17) Students in this class feel like a family.
- (18) Students in this class would be able to resolve conflict if it arises in class.
- (19) Students in this class would give rides to each other if needed.
- (20) A feeling of community spirit exists among the students in this class.
- (21) Students in this class like each other.

- (22) When faced with a problem in this class, students can create a solution.
- (23) Students in this class feel isolated from each other.
- (24) Students in this class share the same values.
- (25) Students in this class influence each others' behavior.
- (26) Students in this class get things done to improve the class.
- (27) Students are committed to the class's success.
- (28) If asked to participate on a class project, students would volunteer willingly.
- (29) Students in this class have a voice regarding important class decisions.
- (30) Students in this class feel that it is a safe place to express their views.
- (31) Students would be willing to work with others on a project and get a common grade.
- (32) Students in this class can persuade the professor to respond to their needs and concerns.
- (33) Students in this class feel they belong here.

Students were briefed about the research at beginning of the semester and they were asked to sign a consent form to participate in the research. The participation of students was voluntary. They took a pre-test at the beginning of the semester and a post-test, which was included as part of the final exam. Each test was composed of thirty questions out of which 20 were multiple choice questions and ten were descriptive questions. Pre-test was intended to measure the prior knowledge of the students and post-

test was intended to measure their knowledge improvement. A statistically significant improvement would indicate effectiveness of the studio-based model.

Exit interviews were conducted at the end of the semester for both courses to obtain feedback from the students. Five students were selected based on their performance in the course, two from the top one percent, one from the middle and two from the bottom one percent. Interview questions were open-ended and addressed student perceptions of learning, interest, motivation, and sense of community with others in the class. Some examples of the interview questions are provided below.

- a. Learning: Did you find the process of completing programming projects helpful to you in learning about computer programming? Why or why not?
- b. Motivation: Did the course keep you interested and motivated to learn? Why or why not?
- c. Community: Are you comfortable giving and receiving feedback on computer programming?

The overall data we obtained showed us that the studio-based model of teaching was effective. A significant positive difference between the pre-test and post-test mean scores was found for each course. Collected data includes pre-test scores, post-test scores and survey scores for the scales described above. The data obtained was analyzed using descriptive statistics, univariate analysis of variance (ANOVA), repeated measures and mixed design. The statistical analyses were conducted at the .05 alpha levels. Univariate analysis of variance (ANOVA) procedures were used to analyze the survey scales and pre-to-post-test scores in both semesters. Repeated measure procedure was used to

analyze the post-to-post test scores across the two semesters and finally the mixed design was used to analyze the pre-to-post test scores across the two semesters.

5.1. CS2: Comparison of Two SBL Implementations

The aim of this analysis is to show the effectiveness (or lack thereof) of the studio model by analyzing the pre-to post-test score difference and pre-to-post survey difference in the selected scales in CS2 during 2007 Fall (make sure to consistently use Fall or Fall, and Spring or spring in the entire thesis) and 2008 spring semesters. The CS2 course was classified as pilot SBL in Fall 07 and as full SBL in Spring 08.

Results of the analysis of pre-test and post-test scores are summarized in Table 5.1. Our main goal is to assess the difference between the pre-test and post-test scores within each semester and across the two semesters. The p-values for both the semesters are less than 0.001 indicating that there is a statistically significant difference (increase) in the scores.

CS2	Pre-Test (Mean)	Post-Test (Mean)	N	F	p
Fall 07 (Pilot SBL)	27.07	54.47	43	3.25	p < 0.001
Spring 08 (SBL)	16.65	40.42	54	7.47	p < 0.001
Computed using alpha = 0.05					

Table 5.1. Summary of pre-to-post test score comparison across two semesters of CS2

A univariate analysis of variance (ANOVA) was performed on the post-test scores of both semesters and the results are summarized in Table 5.2. There is an increase in the mean value across the semesters, but the p value is greater than 0.05. The reason for this may be that the teaching methodology used across both the semesters had studio

components, so there is no significant difference between the post-test means from both semesters.

CS2	Fall 07 (Pilot SBL)	Spring 08 (SBL)
N	43	55
Post-Test (Mean)	54.47	57.00
p = 0.158		
Computed using alpha = 0.05		

Table 5.2. Summary of post-test score comparison across two semesters of CS2

A mixed design ANOVA was performed on the pre-to-post test scores of Fall 07 and Spring 08 (see Table 5.1 for p and F values) to compare the learning trajectories across the two semesters (plotted in Figure 5.1). From the graph we can conclude that the students in Spring 08 started with less knowledge but learned more, on average, than students in the Fall 07 class.

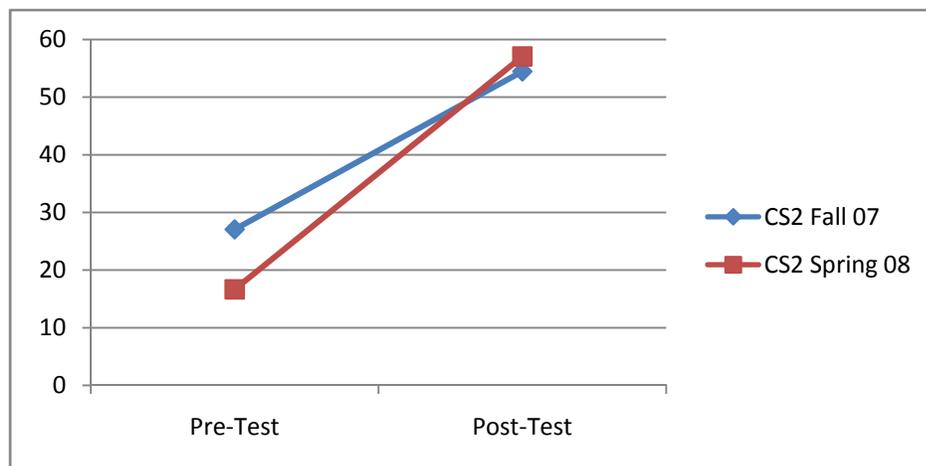


Figure 5.1. Comparison of pre-to-post test scores across the semesters of CS2

The above results indicate that there was a statistically significant effect of the studio-based model of teaching on student learning. This finding answers the first research question.

Repeated Measures analysis of variance (ANOVA) was performed on the pre and post-survey data collected in Fall 07 semester. The results of this analysis are summarized in Table 5.3. Results indicate that the level of intrinsic motivation ($p < 0.001$), efficacy ($p < 0.001$), critical thinking ($p = 0.002$) and self regulation ($p = 0.019$) significantly increased from pre-survey to the post-survey during Fall 07 with p values < 0.05 .

CS2 –Fall 07 (Pilot SBL)	Pre-Survey (Mean)	Post-Survey (Mean)	F(1,40)	p
Intrinsic Motivation	5.00	6.01	52.14	p < 0.001
Extrinsic Motivation	5.03	5.16	0.415	p = 0.523
Efficacy	4.67	5.46	18.82	p < 0.001
Peer Learning	3.65	3.70	0.023	p = 0.881
Critical Thinking	4.32	4.85	10.26	p = 0.002
Self Regulation	4.28	4.60	5.94	p = 0.019
Sense of Community	2.66	2.72	0.210	p = 0.650
N=40				
Computed using alpha = 0.05				

Table 5.3. Summary of pre-to-post survey score comparison for Fall 07 semester of CS2

CS2 –Spring08 (SBL)	Pre-Survey (Mean)	Post-Survey (Mean)	F(1,45)	p
Intrinsic Motivation	5.06	5.57	52.14	p < 0.001
Extrinsic Motivation	5.16	5.54	4.65	p = 0.036
Efficacy	5.11	5.92	22.27	p < 0.001
Peer Learning	3.06	3.89	12.68	p < 0.001
Critical Thinking	4.11	4.16	0.064	p = 0.801
Self Regulation	4.47	5.26	41.02	p < 0.001
Sense of Community	2.05	2.82	48.37	p < 0.001
N=45				
Computed using alpha = 0.05				

Table 5.4. Summary of pre-to-post survey score comparison for Fall 07 semester of CS2

Repeated measures analysis of variance (ANOVA) was performed on the pre and post-survey data collected in Spring 08 semester. The results of this analysis are summarized in Table 5.4, and indicate that the level of intrinsic motivation ($p < 0.001$), extrinsic motivation ($p = 0.036$), efficacy ($p < 0.001$), peer learning ($p < 0.001$), self regulation ($p = 0.019$) and sense of community ($p < 0.001$) significantly increased from pre-survey to the post-survey with all the scales having $p\text{-value} < 0.05$.

A univariate analysis of variance (ANOVA) was performed on the post to post-survey data collected in Fall 07 and Spring 08 semesters. The results of this analysis are summarized in Table 5.5, and indicate that the level of intrinsic motivation decreased ($p < 0.001$), but levels of extrinsic motivation ($p = 0.028$), efficacy ($p < 0.001$), and self regulation ($p = 0.019$) were higher in Spring 08 when compared to Fall 07.

Post-Survey CS2	Fall 07 (Pilot SBL)	Spring 08 (SBL)	F(1,85)	p
N	40	45	-	-
Intrinsic Motivation	6.01	5.57	12.53	p < 0.001
Extrinsic Motivation	5.16	5.54	4.97	p = 0.028
Efficacy	5.46	5.92	13.99	p < 0.001
Peer Learning	3.70	3.89	0.79	p = 0.374
Critical Thinking	4.85	4.16	0.162	p = 0.688
Self Regulation	4.60	5.26	22.84	p < 0.001
Sense of Community	2.72	2.82	0.048	p = 0.828
Computed using alpha = 0.05				

Table 5.5. Summary of post-to-post survey score comparison for Fall 07 semester and Spring 08 semester of CS2

Mixed design ANOVAs were performed on the pre-to-post survey data of Fall 07 and Spring 08 to compare the pre to post change across the two semesters. The output is plotted in Figures 5.2 - 5.8. It can be seen that the levels of all seven scales improved from pre- to post-survey in both semesters. The full SBL implementation of studios in spring 2008 resulted in greater improvements in the scales of extrinsic motivation, efficacy, peer learning, self regulation and sense of community. Increases in the scale levels of intrinsic motivation and critical thinking were more in the Fall 07 pilot studio implementation.

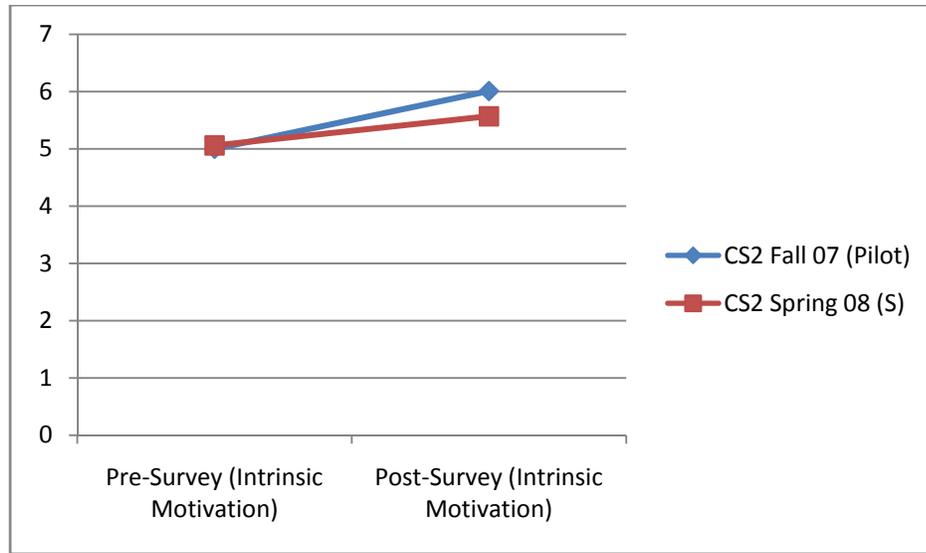


Figure 5.2. Comparison of pre-to-post survey scores of intrinsic motivation across two semesters of CS2

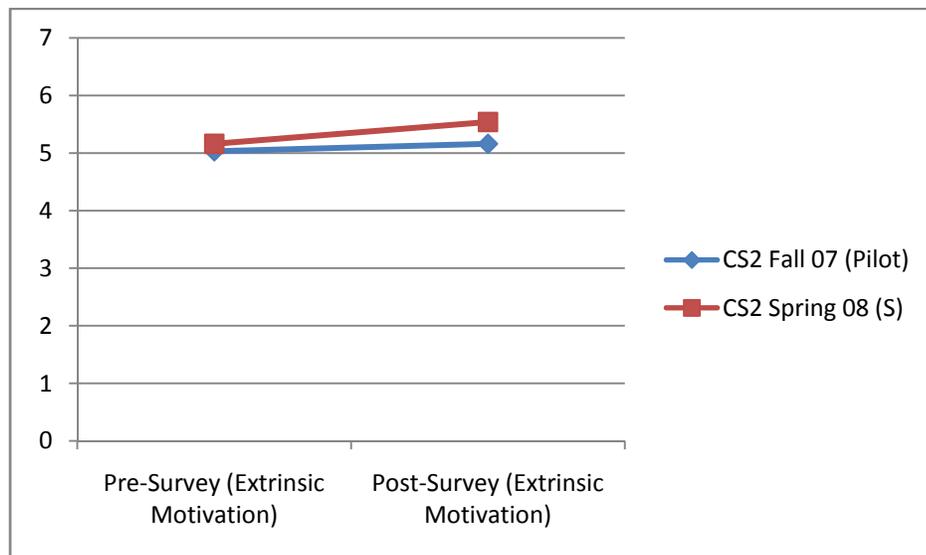


Figure 5.3. Comparison of pre-to-post survey scores of extrinsic motivation across two semesters of CS2

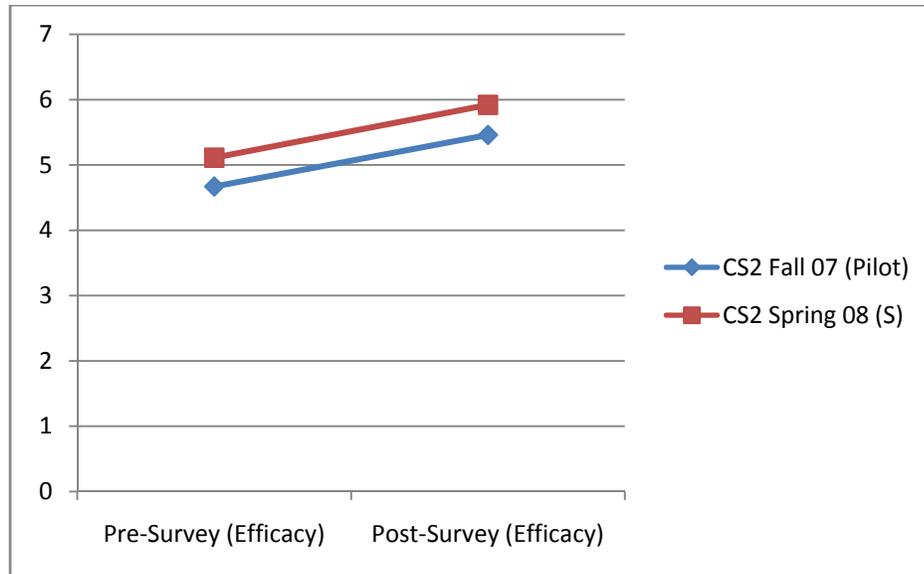


Figure 5.4. Comparison of pre-to-post survey scores of efficacy across two semesters of CS2

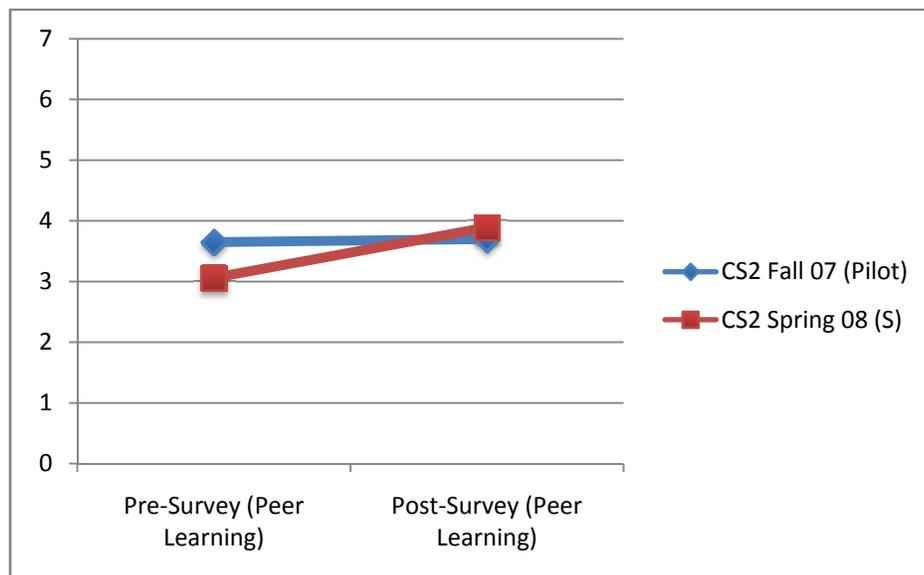


Figure 5.5. Comparison of pre-to-post survey scores of peer learning across two semesters of CS2

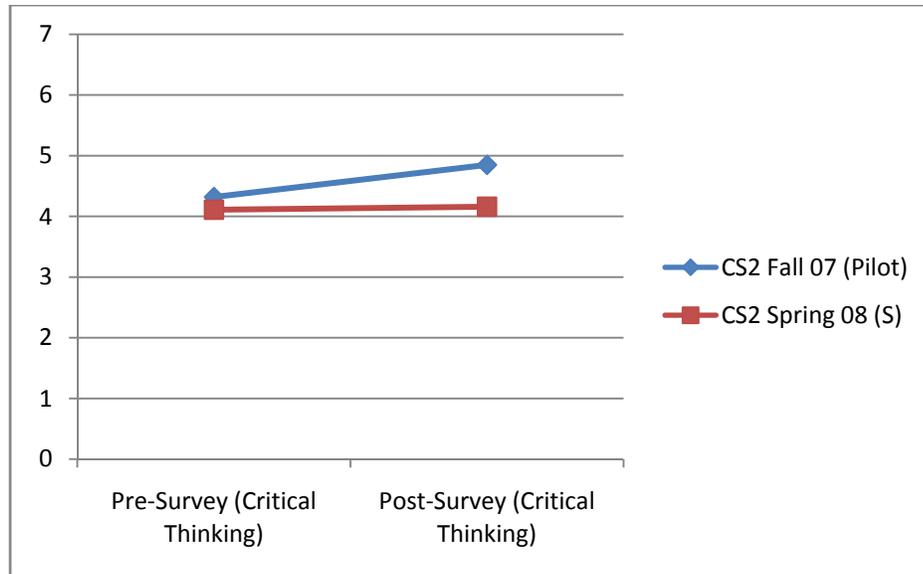


Figure 5.6. Comparison of pre-to-post survey scores of critical thinking across two semesters of CS2

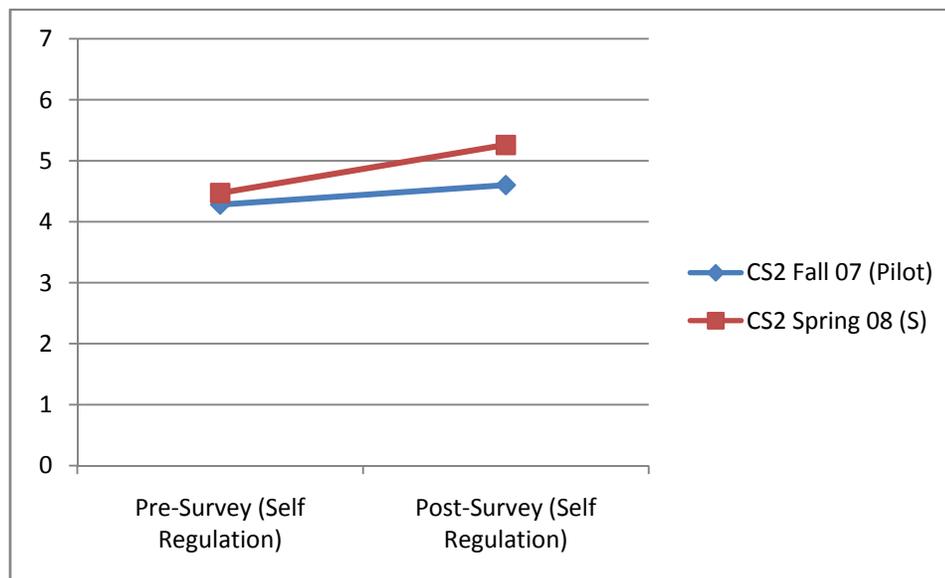


Figure 5.7. Comparison of pre-to-post survey scores of self regulation across two semesters of CS2

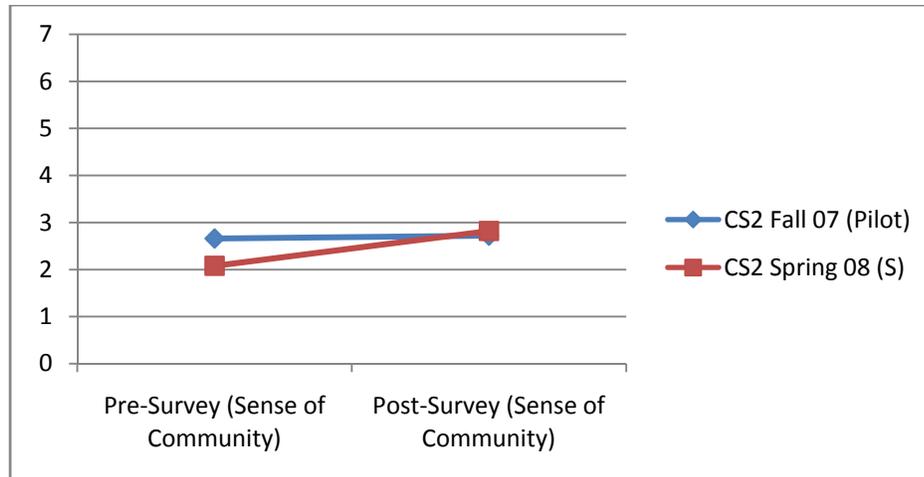


Figure 5.8. Comparison of pre-to-post survey scores of sense of community across two semesters of CS2

The analysis of student performance data showed that students learned better in the full-fledged studio implementation compared to the pilot studio implementation of CS2. The analyses of student attitude and motivation survey data showed that students in the full-fledged studio implementation of CS2 emerged with higher sense of community, extrinsic motivation, self regulation, efficacy and peer learning compared to the pilot studio implementation.

5.2. CS3: Comparison of Two SBL Implementations

The CS3 course was offered in the traditional format in Fall 07 and in the studio-based format in Spring 08. An identical set of analyses were carried out on the data collected from CS3 students in these two semesters.

The results computed on pre-test and post-test scores are summarized in Table 5.6. The p-values for both the semesters are less than 0.001. This means that there is a statistically significant difference (increase) in student scores from pre-test to post-test in both semesters.

CS3	Pre-Test (Mean)	Post-Test (Mean)	N	p
Fall 07 (Traditional)	19	44.26	28	p < 0.001
Spring 08 (SBL)	29	50.71	29	p < 0.001

Table 5.6. Summary of pre-to-post test scores comparison in the two semesters in CS3

A univariate analysis of variance (ANOVA) was performed on the post-test scores of both semesters and the results are summarized in Table 5.7. The mean value on the post test was higher in the Spring of 08 ($p=0.044$) indicating that the knowledge level of the students was higher at the end of Spring 08 (SBL) when compared to Fall 07 (traditional).

CS3	Fall 07 (Traditional)	Spring 08 (Pilot SBL)
N	28	29
Post-Test (Mean)	44.26	50.71
$p = 0.044$		
Computed using alpha = 0.05		

Table 5.7. Summary of post-test score comparison across two semesters of CS3

A mixed design analysis was performed on the pre-to-post test scores of Fall 07 and Spring 08 (see Table 5.6 for p values) to compare learning trajectories across the two semesters. The results are plotted in Figure 5.9. From the graph we conclude that students in both Fall 07 and Spring 08 did not differ in their knowledge level gains from the beginning to the end of the semesters. This suggests that studio-based instruction had no more effect on the learning of students in the CS3 course than traditional instruction.

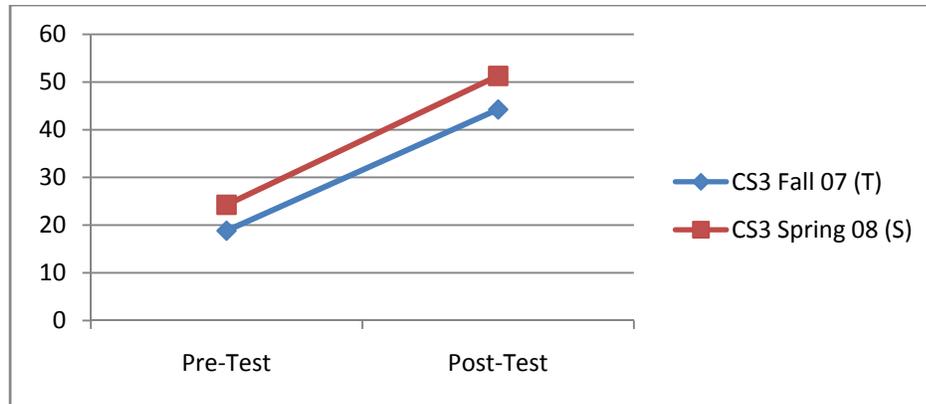


Figure 5.9. Comparison of pre-to-post test scores across two semesters in CS3

Repeated measures analysis of variance (ANOVA) was performed on the pre and post-survey data collected in Fall 07 semester. The results of this analysis are summarized in Table 5.8, and indicate that there were no statistically significant changes in the scales.

CS3 –Fall 07 (Traditional)	Pre-Survey (Mean)	Post-Survey (Mean)	F(1,28)	p
Intrinsic Motivation	5.02	5.16	2.68	p = 0.113
Extrinsic Motivation	4.87	5.25	3.43	p = 0.075
Efficacy	5.32	5.48	1.09	p = 0.305
Peer Learning	3.30	3.72	0.085	p = 0.125
Critical Thinking	4.26	4.37	0.346	p = 0.561
Self Regulation	4.27	4.37	0.326	p = 0.573
Sense of Community	2.77	2.72	7.45	p = 0.658
N=28				
Computed using alpha = 0.05				

Table 5.8. Summary of pre-to-post survey score comparison for Fall 07 semester of CS3

Repeated measures analysis of variance (ANOVA) was performed on the pre and post-survey data collected in Spring 08 semester. The results are summarized in Table 5.9, and indicate that the level of peer learning (p=0.005) and sense of community significantly increased, whereas level of efficacy significantly decreased from pre-survey to post-survey with the scales having p-values < 0.05.

CS3 –Spring08 (SBL)	Pre-Survey (Mean)	Post-Survey (Mean)	F(1,25)	p
Intrinsic Motivation	5.06	4.77	0.351	p = 0.351
Extrinsic Motivation	5.37	5.09	1.84	p = 0.187
Efficacy	5.14	4.61	5.28	p = 0.030
Peer Learning	3.41	4.37	9.67	p = 0.005
Critical Thinking	4.26	4.29	0.235	p = 0.633
Self Regulation	4.27	4.50	1.52	p = 0.229
Sense of Community	2.66	3.53	0.043	p = 0.012
N=25				
Computed using alpha = 0.05				

Table 5.9. Summary of pre-to-post survey score comparison for Spring 08 semester of CS3

A univariate analysis of variance (ANOVA) was performed on the post to post-survey data collected in Fall 07 and Spring 08 semesters. The results of this test are summarized in Table 5.10, which indicate that the level of peer learning (p=0.014) was higher at the end of Spring 08 when compared to the end of Fall 07.

Post-Survey CS3	Fall 07 (Traditional)	Spring 08 (SBL)	F(1,53)	P
N	28	25	-	-
Intrinsic Motivation	5.16	4.77	0.992	p = 0.324
Extrinsic Motivation	5.25	5.09	0.454	p = 0.504
Efficacy	5.48	4.61	3.59	p = 0.064
Peer Learning	3.72	4.37	6.473	p = 0.014
Critical Thinking	4.37	4.29	10.57	p = 0.785
Self Regulation	4.37	4.50	1.85	p = 0.180
Sense of Community	2.72	3.53	0.099	p = 0.755
Computed using alpha = 0.05				

Table 5.10 Summary of post-to-post survey score comparison for Fall 07 semester and Spring 08 semester of CS3

Mixed design ANOVAs were performed on the pre-to-post survey data of Fall 07 and Spring 08 to compare the pre to post values across the two semesters and the output is plotted in Figures 5.10 - 5.17.

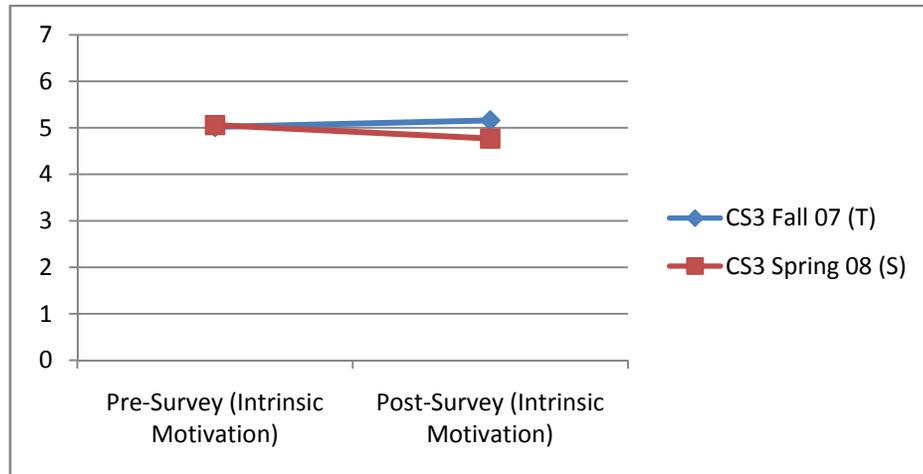


Figure 5.10. Comparison of pre-to-post survey scores of intrinsic motivation across two semesters of CS3

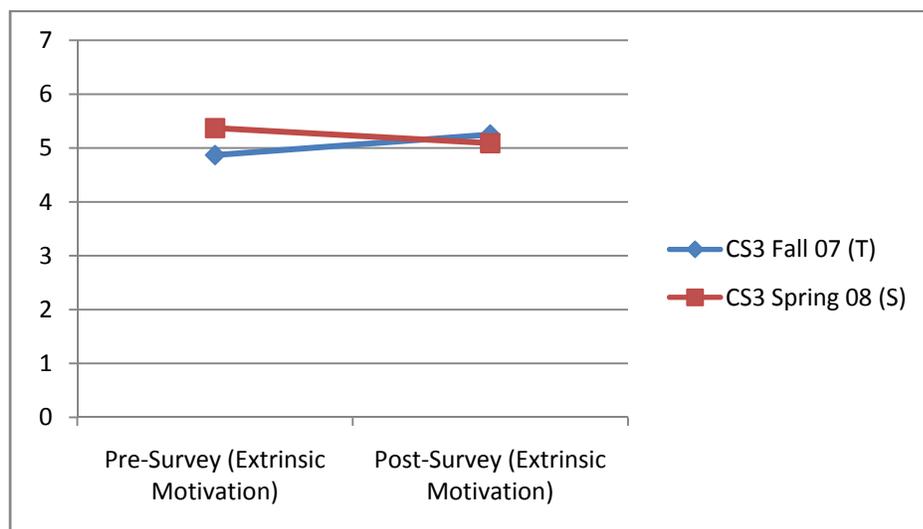


Figure 5.11. Comparison of pre-to-post survey scores of extrinsic motivation across two semesters of CS3

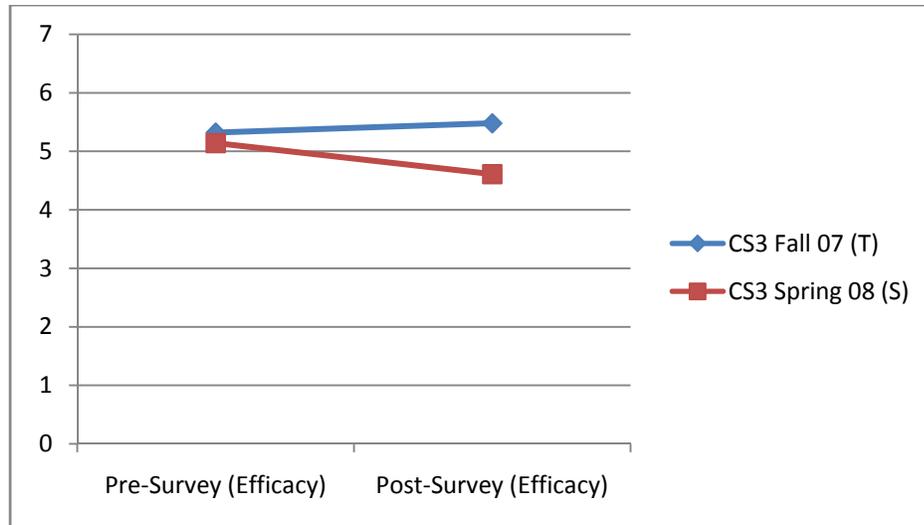


Figure 5.12. Comparison of pre-to-post survey scores of efficacy across two semesters of CS3

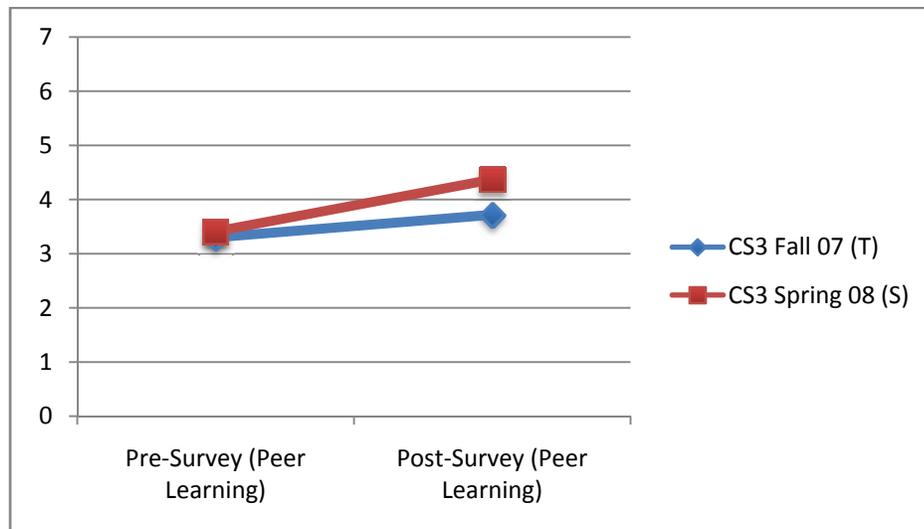


Figure 5.13. Comparison of pre-to-post survey scores of peer learning across two semesters of CS3

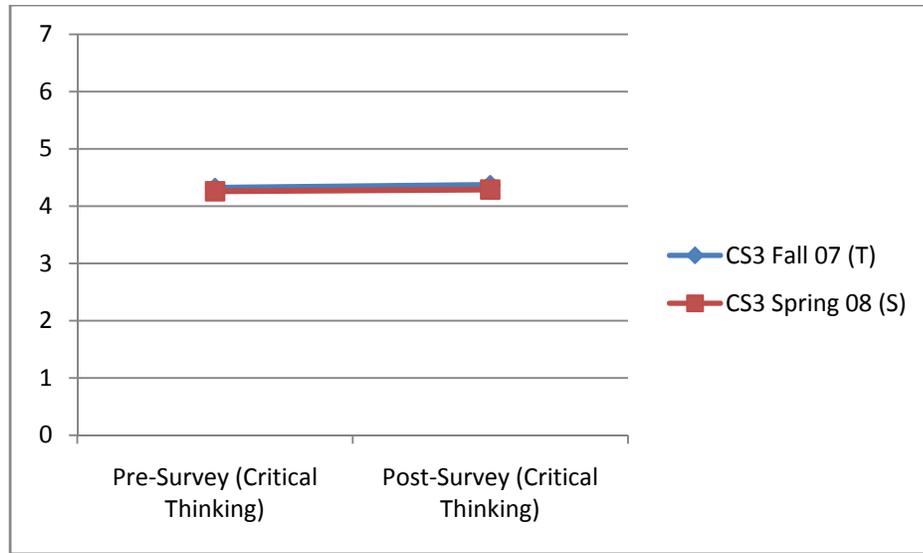


Figure 5.14. Comparison of pre-to-post survey scores of critical thinking across two semesters of CS3

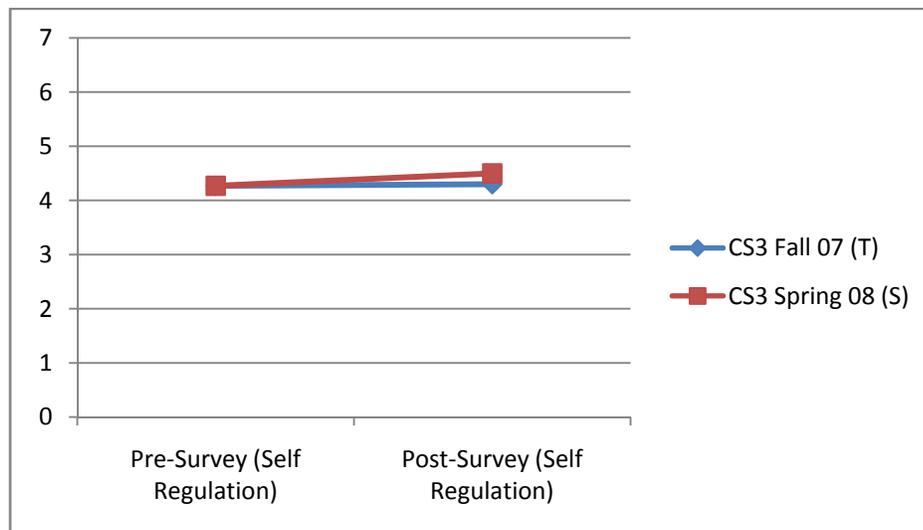


Figure 5.15. Comparison of pre-to-post survey scores of self regulation across two semesters of CS3

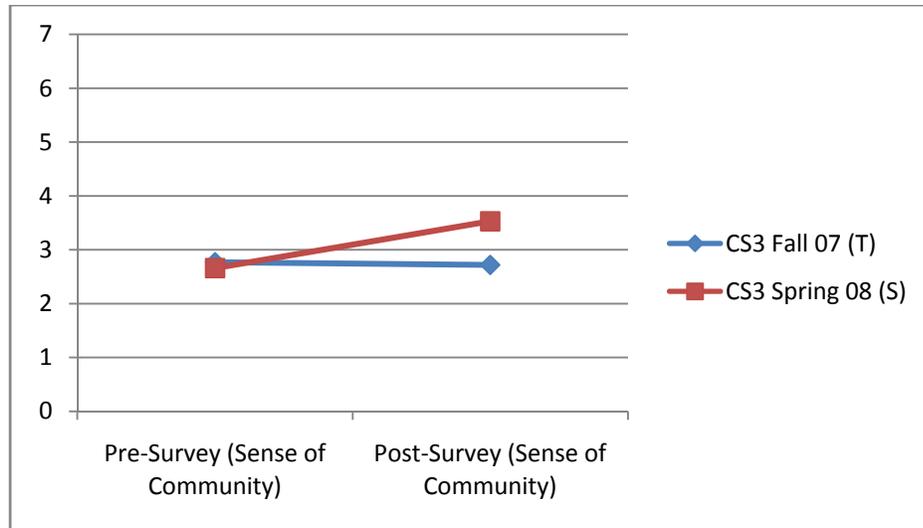


Figure 5.16. Comparison of pre-to-post survey scores of sense of community across two semesters of CS3

The analysis of student performance data comparing studio and traditional implementation in CS3 did not yield a statistically significant difference. The figures above indicate that there were no between-semester differences in pre- to post- changes of the levels of self regulation and critical thinking. Peer learning increased more in the pilot studio implementation, but levels of intrinsic and extrinsic motivation, sense of community and efficacy were lower in the studio semester. These are unexpected results, contrary to the results we obtained in CS2. Possible reasons for this and future research directions are discussed in the next chapter, which concludes this thesis.

CHAPTER 6

CONCLUSION AND FUTURE RESEARCH

In this thesis we addressed the problem of motivating and engaging undergraduate students of computer science, and enabling them to better learn data structures and algorithms, by implementing and evaluating a novel pedagogical approach called studio-based learning. Studio-based learning is an instructional technique that emphasizes collaborative, design-oriented learning. The two key activities in this approach are the following. 1) Students collaboratively use various techniques to develop solutions to complex problems. 2) Students present their solutions to their instructors and peers for feedback and discussion. We developed three support systems – Peer Review System, Survey Data Collection System, and Solution Uploading system – required for the implementation of studio-based learning approach in the classrooms. A web-based portal was also designed that enables the studio-based learning community to communicate and collaborate on a regular basis.

We implemented the studio-based learning approach in two computer science courses (CS2 and CS3) at Auburn University. In Fall 07 semester we offered the CS2 course in pilot studio-based learning format and CS3 in traditional format. In Spring 08, CS2 was offered in full studio format and CS3 was offered in pilot studio format. The pilot studio implementation of CS2 revealed some problems, and we made many design changes to its SBL implementation for Spring 08 semester to address these problems.

Different types of data (pre-test, post-test, and pre- and post-surveys of attitude and sense of community) were collected from both studio and traditional implementations of the two courses to analyze and study the effectiveness of the studio-based learning approach as compared to the traditional approach.

The analyses done on pre- and post-test scores suggest that students' learning gain was higher in the full studio implementation of CS2 when compared to the pilot studio implementation, whereas there was no difference in student learning gains in the case of CS3 between traditional and pilot studio implementations.

The analyses done on the survey data indicate that the students in CS2 reported higher levels of sense of community, extrinsic motivation, efficacy, self regulation and peer learning in Spring 08 full studio semester when compared to Fall 07 partial studio semester, whereas in CS3 the students exhibited better peer learning and sense of community in the pilot studio of Spring 08 but their levels of intrinsic and extrinsic motivation and efficacy decreased from Fall 07 to Spring 08.

There are many possible reasons for the mixed results we obtained in CS3 –there was no difference in student learning gains between traditional and studio formats, and the seven scales measured in surveys varied (two increased, three decreased and two stayed about the same). Perhaps studio-based learning is more suited to a design and programming oriented course such as CS2 than a mathematical and analytical course like CS3. Alternately, as CS3 did not have separate lab sessions, studios took away 15% of the available lecture time, as a result of which the lectures had to be compressed to cover the same curriculum in the pilot studio implementation as that of the traditional

implementation. This may have offset any benefits of studios. Further research is needed to explore this issue.

In our future work we will further investigate the impact of studios in CS3, extend the implementation of the studio-based learning approach beyond the current two courses, and we will compare our evaluation and results across different universities implementing studio-based learning.

REFERENCES

1. Begel, A., and Simon, B. 2008. Struggles of new college graduates in their first software development job. *Proceedings of 39th Technical Symposium on Computer Science Education (SIGCSE 2008)*, March 12-15, 2008, Portland, Oregon, pp. 226-230.
2. Hundhausen, C. D., Narayanan, N. H., and Crosby, M. E. (2008). Exploring studio-based instructional models for computing education. *Proceedings of the SIGCSE Technical Symposium on Computer Science Education*, ACM Press, pp. 392-396.
3. Myneni, L., Ross, M., Hendrix, D., and Narayanan, N. H. (2008). Studio-based learning in CS2: An experience report. *Proceedings of the 46th Annual ACM Southeast Conference*, ACM Press, pp. 253-255.
4. Hendrix D., Myneni, L., Narayanan, N. H., and Ross, M. (2008). Adapting a studio-based learning model for CS2. *Technical Report CSSE08-03, Computer Science & Software Engineering Dept.*, Auburn University, Auburn, AL 36849.
5. Hundhausen, C.D., and Brown, J.L. (2008). Designing, visualizing, and discussing algorithms within a CS 1 studio experience: An empirical study. *Computers & Education 50(1)*, pp. 301-326.
6. Hundhausen, C.D., and Brown, J.L. (2007). What you see is what you code: A 'live' algorithm development and visualization environment for novice learners. *Journal of Visual Languages and Computing 18(1)*, pp. 22-47.
7. Hübscher-Younger, T., and Narayanan, N. H. (2007). Turning the tables: Investigating characteristics and efficacy of student-authored multimedia representations. Chapter in R. Lowe & W. Schnotz (Eds.), *Learning with Animations Research Implications for Design*, Cambridge University Press, pp.235-262.
8. Hübscher-Younger, T., and Narayanan, N. H. (2003). Constructive and collaborative learning of algorithms. *Proceedings of the SIGCSE Technical Symposium on Computer Science Education*, ACM Press, pp. 6-10.

9. Hübscher-Younger, T., and Narayanan, N. H. (2003). Dancing hamsters and marble statues: Characterizing student visualizations of algorithms. *Proceedings of the First ACM Symposium on Software Visualization*, ACM Federated Computing Research Conference, ACM Press, pp. 95-104.
10. D'Amico, J., Hübscher-Younger, T., Hübscher, R., and Narayanan, N. H. (2003). Applying contextual design to build a course scheduler: A case study. *Proceedings of the 41st Annual ACM Southeast Regional Conference*, ACM Press, pp. 330-336.
11. Hübscher-Younger, T., and Narayanan, N. H. (2003). Designing for divergence. B. Wasson, S. Ludvigsen & U. Hoppe, (Eds.), *Designing for Change in Networked Learning Environments: Proceedings of the International Conference on Computer Support for Collaborative Learning*, pp. 461-470, Kluwer Academic Publishers.
12. Hansen, S. R., Narayanan, N. H., and Hegarty, M. (2002). Designing educationally effective algorithm visualizations: Embedding analogies and animations in hypermedia. *Journal of Visual Languages and Computing*, 13(3): 291-317, Academic Press.
13. Hundhausen, C.D. (2002). Integrating algorithm visualization technology into an undergraduate algorithms course: Ethnographic studies of a social constructivist approach. *Computers & Education* 39 (3), 237-260.
14. Carbone, A., and Sheard, J. 2002. A studio-based teaching and learning model in IT: What do first year students think? *Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education*, Aarhus, Denmark, 2002, pp. 213-217.
15. Hundhausen, C.D, and Douglas, S.A. (2000). SALSA and ALVIS: A language and system for constructing and presenting low fidelity algorithm visualizations. *Proceedings of the 2000 IEEE Symposium on Visual Languages* (pp. 67-68). Los Alamitos, CA: IEEE Computer Society Press.
16. Hundhausen, C. (2002). The "Algorithms Studio" project: Using sketch-based visualization technology to construct and discuss visual representations of algorithms. *Proceedings of the 2002 IEEE Symposia on Human Centric Computing Languages and Environments* (pp. 99-100). Los Alamitos, CA: IEEE Computer Society Press.
17. Docherty, M., Sutton, P., Brereton, M., and Kaplan, S. 2001. An innovative design and studio-based CS degree. *ACM SIGCSE Bulletin*, Vol. 33, Issue 1, March 2001, pp. 233-237.

18. Pintrich, P.R., Smith, D. A. F., Garcia, T., and McKeachie, W. J. 1991. A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ). National Center for Research to Improve Postsecondary Teaching and Learning.
19. Hundhausen, C. D., Narayanan, N. H., and Crosby, M. E. (2007). Collaborative research: CPATH CB: Exploring studio-based instructional models for computing education. .Project proposal submitted to Nation Science Foundation, January 2007.
20. Schon, D. 1983. The Reflective Practitioner: How Professionals Think in Action. New York: Basic Books.

APPENDIX A

MOTIVATED STRATEGY LEARNING QUESTIONNAIRE (MSLQ)

SURVEY INSTRUMENT

DEMOGRAPHIC INFORMATION

(All information collected will be kept strictly confidential and will not be associated with your name, per the informed consent agreement)

1. Name: _____

2. Gender (circle one) ___ Male ___ Female

3. Age (circle one) ___ 18 ___ 19-20 ___ 21-22 ___ 23-24 ___ 25+

4. What is your academic Major? _____ Minor? _____

5. Class level (circle one)

___ Freshman ___ Sophomore ___ Junior ___ Senior ___ Graduate

6. Ethnic Background

___ African-American ___ Asian-American ___ Caucasian ___ Hispanic or Spanish Speaking ___ Other

7. How many courses have you taken in engineering?

___ 0 ___ 1-2 ___ 3-5 ___ 6-8 ___ 9+

8. How many hours per week do you study engineering subjects?

___ 0 ___ 1-5 ___ 6-10 ___ 11-20 ___ 21+

9. How many hours per week do you prepare for this engineering course?

___ 0 ___ 1-5 ___ 6-10 ___ 11-20 ___ 21+

10. How important were the following for your decision to take this course/class.

(1 = not at all important, 5 = very important)

a. fulfills a course requirement	1	2	3	4	5
b. experience seemed interesting	1	2	3	4	5
c. is required	1	2	3	4	5
d. will be useful to me in school	1	2	3	4	5
e. will be useful to me in life	1	2	3	4	5
f. will help improve my academic skills	1	2	3	4	5
g. was recommended by a friend	1	2	3	4	5
h. was recommended by an advisor/professor	1	2	3	4	5
i. will improve career prospects	1	2	3	4	5
j. fit into my schedule	1	2	3	4	5

11. How confident are you in the following

(1 = not at all confident, 5 = very confident)

a. engineering	1	2	3	4	5
b. math	1	2	3	4	5
c. reading	1	2	3	4	5
d. writing	1	2	3	4	5
e. science	1	2	3	4	5
f. computer science	1	2	3	4	5

MSLQ- Part A. Motivation

The following questions ask about your motivation for and attitudes about the class. **Remember there is no right or wrong answers; just please answer as accurately as possible.** Use the scale below to answer the questions. If you think the statement is very true of you, circle 7; if a statement is not at all true of you, circle 1. If the statement is more or less true of you, find a number between 1 and 7 that best describes you.

	1	2	3	4	5	6	7			
	not at all true of me						very true of me			
				not at all true of me			very true of me			
1.				1	2	3	4	5	6	7
	In a class like this, I prefer course material that really challenges me so I can learn new things.									
2.				1	2	3	4	5	6	7
	If I study in appropriate ways, then I will be able to learn the material in this course.									
3.				1	2	3	4	5	6	7
	When I take a test I think about how poorly I am doing compared to other students.									
4.				1	2	3	4	5	6	7
	I think I will be able to use what I learn in this course in other courses.									
5.				1	2	3	4	5	6	7
	I believe I will receive an excellent grade in this class.									
6.				1	2	3	4	5	6	7
	I'm certain I can understand the most difficult material presented in the reading for this course.									
7.				1	2	3	4	5	6	7
	Getting a good grade in this class is the most satisfying thing for me right now.									
8.				1	2	3	4	5	6	7
	When I take a test I think about items on other parts of the test I can't answer.									
9.				1	2	3	4	5	6	7
	It is my own fault if I don't learn the material in this course.									
10.				1	2	3	4	5	6	7
	It is important for me to learn the course material in this class.									

	not at all true of me						very true of me
11. The most important thing for me right now is improving my overall grade point average, so my main concern in this class is getting a good grade.	1	2	3	4	5	6	7
12. I'm confident I can learn the basic concepts taught in this course.	1	2	3	4	5	6	7
13. If I can, I want to get better grades in this class than most of the other students.	1	2	3	4	5	6	7
14. When I take tests I think of the consequences of failing.	1	2	3	4	5	6	7
15. I'm confident I can understand the most complex material presented by the instructor in this course.	1	2	3	4	5	6	7
16. In a class like this, I prefer course material that arouses my curiosity, even if it is difficult to learn.	1	2	3	4	5	6	7
17. I am very interested in the content area of this course.	1	2	3	4	5	6	7
18. If I try hard enough, then I will understand the course material.	1	2	3	4	5	6	7
19. I have an uneasy, upset feeling when I take an exam.	1	2	3	4	5	6	7
20. I'm confident I can do an excellent job on the assignments and tests in this course.	1	2	3	4	5	6	7
21. I expect to do well in this class.	1	2	3	4	5	6	7
22. The most satisfying thing for me in this course is trying to understand the content as thoroughly as possible.	1	2	3	4	5	6	7
23. I think the course material in this class is useful for me to learn.	1	2	3	4	5	6	7

	not at all true of me						very true of me
24. 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.	1	2	3	4	5	6	7
25. If I don't understand the course material, it is because I didn't try hard enough.	1	2	3	4	5	6	7
26. I like the subject matter of this course.	1	2	3	4	5	6	7
27. Understanding the subject matter of this course is very important to me.	1	2	3	4	5	6	7
28. I feel my heart beating fast when I take an exam.	1	2	3	4	5	6	7
29. I'm certain I can master the skills being taught in this class.	1	2	3	4	5	6	7
30. I want to do well in this class because it is important to show my ability to my family, friends, employer, or others.	1	2	3	4	5	6	7
31. Considering the difficulty of this course, the teacher, and my skills, I think I will do well in this class.	1	2	3	4	5	6	7

MSLQ - Part B. Learning Strategies

The following questions ask about your learning strategies and study skills. **Again, there is no right or wrong answers. Please answer the questions about how you study as accurately as possible.** Use the same scale to answer the remaining questions. If you think the statement is very true of you, circle 7; if a statement is not at all true of you, circle 1. If the statement is more or less true of you, find a number between 1 and 7 that best describes you.

	1	2	3	4	5	6	7
	not at all true of me						very true of me
32. When I study the reading for this course, I outline the material to help me organize my thoughts.	1	2	3	4	5	6	7
33. During class time I often miss important points because I'm thinking of other things.	1	2	3	4	5	6	7
34. When studying for this course, I often try to explain the material to a classmate or friend.	1	2	3	4	5	6	7
35. I usually study in a place where I can concentrate on my course work.	1	2	3	4	5	6	7
36. When reading for this course, I make up questions to help focus my reading.	1	2	3	4	5	6	7
37. I often feel so lazy or bored when I study for this class that I quit before I finish what I planned to do.	1	2	3	4	5	6	7
38. I often find myself questioning things I hear or read in this course to decide if I find them convincing.	1	2	3	4	5	6	7
39. When I study for this class, I practice saying the material to myself over and over.	1	2	3	4	5	6	7
40. Even if I have trouble learning the material in this class, I try to do the work on my own, without help from anyone.	1	2	3	4	5	6	7

	not at all true of me						very true of me
41. When I become confused about something I'm reading for this class, I go back and try to figure it out.	1	2	3	4	5	6	7
42. When I study for this course, I go through the readings and my class notes and try to find the most important ideas.	1	2	3	4	5	6	7
43. I make good use of my study time for this course.	1	2	3	4	5	6	7
44. If course readings are difficult to understand, I change the way I read the material.	1	2	3	4	5	6	7
45. I try to work with other students from this class to complete the course assignments.	1	2	3	4	5	6	7
46. When studying for this course, I read my class notes and the course readings over and over again.	1	2	3	4	5	6	7
47. When a theory, interpretation, or conclusion is presented in class or the readings, I try to decide if there is good supporting evidence.	1	2	3	4	5	6	7
48. I work hard to do well in this class even if I don't like what we are doing.	1	2	3	4	5	6	7
49. I make simple charts, diagrams, or tables to help me organize course material.	1	2	3	4	5	6	7
50. When studying for this course, I often set aside time to discuss course material with a group of students from the class.	1	2	3	4	5	6	7
51. I treat the course material as a starting point and try to develop my own ideas about it.	1	2	3	4	5	6	7
52. I find it hard to stick to a study schedule.							

	not at all true of me						very true of m
53. When I study for this course, I often set aside time to discuss course material with a group of students from the class.	1	2	3	4	5	6	7
54. Before I study new course material thoroughly, I often skim it to see how it is organized.	1	2	3	4	5	6	7
55. I ask myself questions to make sure I understand the material I have been studying in this class.	1	2	3	4	5	6	7
56. I try to change the way I study in order to fit the course requirements and the instructor's teaching style.	1	2	3	4	5	6	7
57. I often find that I have been reading for this class but don't know what it was all about.	1	2	3	4	5	6	7
58. I ask the instructor to clarify concepts I don't understand well.	1	2	3	4	5	6	7
59. I memorize key words to remind me of important concepts in this class.	1	2	3	4	5	6	7
60. When course work is difficult, I either give up or only study the easy parts.	1	2	3	4	5	6	7
61. I try to think through a topic and decide what I am supposed to learn from it rather than just reading it over when studying for this course.	1	2	3	4	5	6	7
62. I try to relate ideas in this subject to those in other courses whenever possible.	1	2	3	4	5	6	7
63. When I study for this course, I go over my class notes and make an outline of important concepts.	1	2	3	4	5	6	7
64. When reading for this class, I try to relate the material to what I already know.	1	2	3	4	5	6	7

	not at all true of me						very true of me
65. I have a regular place set aside for studying.							
66. I try to play around with ideas of my own related to what I am learning in this course.	1	2	3	4	5	6	7
	1	2	3	4	5	6	7
67. When I study for this course, I write brief summaries of the main ideas from the readings and my class notes.	1	2	3	4	5	6	7
68. When I can't understand the material in this course, I ask another student in this class for help.	1	2	3	4	5	6	7
69. I try to understand the material in this class by making connections between the readings and the concepts from the lectures.	1	2	3	4	5	6	7
70. I make sure that I keep up with the weekly readings and assignments for this course.	1	2	3	4	5	6	7
71. Whenever I read or hear an assertion or conclusion in this class, I think about possible alternatives.	1	2	3	4	5	6	7
72. I make lists of important items for this course and memorize the lists.	1	2	3	4	5	6	7
73. I attend this class regularly.	1	2	3	4	5	6	7
74. Even when course materials are dull and uninteresting, I manage to keep working until I finish.	1	2	3	4	5	6	7
75. I try to identify students in this class whom I can ask for help if necessary.	1	2	3	4	5	6	7
76. When studying for this course I try to determine which concepts I don't understand well.	1	2	3	4	5	6	7

- | | not at all
true of me | | | | | | very true
of me |
|--|--------------------------|---|---|---|---|---|--------------------|
| 77. I often find that I don't spend very much time on this course because of other activities. | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 78. When I study for this class, I set goals for myself in order to direct my activities in each study period. | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 79. If I get confused taking notes in class, I make sure I sort it out afterwards. | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 80. I rarely find time to review my notes of readings before an exam. | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 81. I try to apply ideas from course readings in other class activities such as lecture and discussion. | | | | | | | |

Sense of Community Questionnaire

DIRECTIONS: *Using the following categories, rate the statements below:*

- (1). *Strongly agree or definitely true*
- (2). *Agree or true*
- (3). *Neutral, not sure, or don't know*
- (4). *Disagree or not true*
- (5). *Strongly disagree or definitely not true*

1. Students in this class know each other.	1	2	3	4	5
2. Students in this class work together (e.g., in projects, term papers, etc.)	1	2	3	4	5
3. Students in this class socialize with each other before and after class	1	2	3	4	5
4. A student can feel comfortable answering the teachers questions in this class.	1	2	3	4	5
5. In this class students could talk with each other about problems with the class.	1	2	3	4	5
6. Students in this class help one another, e.g., taking notes if one has to be absent, etc.	1	2	3	4	5
7. Students in this class feel connected to each other.	1	2	3	4	5
8. Students in this class feel comfortable borrowing and lending things from each other (books, notes, etc.)	1	2	3	4	5
9. Students in this class think of themselves as a community.	1	2	3	4	5
10. The students in this class help each other out when someone has a problem.	1	2	3	4	5
11. Students in this class do things together outside of class.	1	2	3	4	5
12. Students in this class support one another.	1	2	3	4	5
13. Students in this class can make it a better course.	1	2	3	4	5
14. Students in this class trust each other.	1	2	3	4	5
15. Students in this class would comfort each other when someone does poorly (e.g., on a test.)	1	2	3	4	5
16. Students in this class hang out together on campus.	1	2	3	4	5
17. Students in this class feel like a family.	1	2	3	4	5
18. Students in this class would be able to resolve conflict if it arises in class.	1	2	3	4	5
19. Students in this class would give rides to each other if needed.	1	2	3	4	5
20. A feeling of community spirit exists among the students in this class.	1	2	3	4	5
21. Students in this class like each other.	1	2	3	4	5
22. When faced with a problem in this class, students can create a solution.	1	2	3	4	5
23. Students in this class feel isolated from each other.	1	2	3	4	5
24. Students in this class share the same values.	1	2	3	4	5
25. Students in this class influence each others' behavior.	1	2	3	4	5
26. Students in this class get things done to improve the class.	1	2	3	4	5
27. Students are committed to the class's success.	1	2	3	4	5
28. If asked to participate on a class project, students would volunteer willingly.	1	2	3	4	5
29. Students in this class have a voice regarding important class decisions.	1	2	3	4	5
30. Students in this class feel that it is a safe place to express their views.	1	2	3	4	5
31. Students would be willing to work with others on a project and get a common grade.	1	2	3	4	5
32. Students in this class can persuade the professor to respond to their needs and concerns.	1	2	3	4	5
33. Students in this class feel they belong here.	1	2	3	4	5

McKinney, J.P., McKinney, K., Franiuk, R., & Schweitzer, J. (2006). The college classroom as a community: Impact on student attitudes and learning. *College Teaching*, 54, 281-284.

➤ 34. Overall, what is your feeling about the sense of community in this class?

_____ Too much → **What should be changed?**

_____ Just about right

_____ Too little → **How could it be increased?**

➤35. What is it that you feel contributes most to your sense of community in your class?

➤36. What is the best thing about being in this class?

➤37. What is the major problem facing this class?

APPENDIX B
CS2 PRE TEST

Name: _____

**COMP 2210
Spring 2009**

Course Pretest

The following pretest will help us assess your current knowledge of computing concepts and skills that will be covered in this course, so that we can better tailor this course to your needs. You have **45 minutes** to complete this test. If you finish early, please bring your test to the proctor. For multiple choice questions, write the letter (A,B,C,D or E) of the correct answer on the space to the left of each question.

Important:

Please do not worry if you are unable to answer all of the questions.

Questions differ in their difficulty. Instead answering them in the given order, do the easy ones first.

How well you do on this test will not impact your course grade; however, you are required to complete this pretest as part of the course.

Please begin this test only when you are instructed to do so.

Use this page for scratch work

E. none of the above

_____ 7. If the array of integers [5, 12, 3, 1, 16] is passed as the input to bubble sort, at the end of the **first** pass through the array, it will look like:

- A. [12, 5, 3, 1, 16]
- B. [5, 12, 3, 1, 16]
- C. [1, 3, 5, 12, 16]
- D. [5, 1, 3, 12, 16]
- E. none of the above

_____ 8. Which of the following statements are true?

- A. Binary Search can be implemented in $O(\log n)$ time using an array
- B. Binary Search can be implemented in $O(\log n)$ time using a singly linked list
- C. Binary Search can be implemented in $O(\log n)$ time using a doubly linked list
- D. All three statements above are true
- E. none of the above statements is true

_____ 9. Which of the following statements about a linked list is **false**?

- A. It is implemented with pointers
- B. Time needed to access any element or node in the list is the same
- C. It has no predetermined or fixed size
- D. Its nodes can be linked in the forward and backward directions
- E. none of the above

_____ 10. How can you use a stack to reverse a series of numbers?

- A. Push a number, pop it, and then repeat this for each of the other numbers
- B. Push two numbers at a time, pop both, then repeat this for the next pair of numbers and so on
- C. Push all the numbers first, then pop all of them
- D. Push first half of the input numbers, pop all of them, then push the second half and pop those
- E. none of the above

_____ 11. If a program takes a sequence of integers, constructs a min heap from them, and then removes and prints the root of the heap and reorders the remaining heap again and again until the heap is empty, how will the integers be printed out?

- A. From the smallest to the largest
- B. In the reverse order as they were inputted into this program
- C. In the same order as they were inputted into this program
- D. From the largest to the smallest
- E. none of the above

_____ 12. What is a hashing function?

- A. A function that implements an arithmetic operation
- B. A function used to iterate through a list
- C. A function that generates a random number
- D. A function that maps an element or key to its location in a table
- E. none of the above

_____ 13. Which of the following will **never** happen if the chaining method, with each table cell pointing to a linked list of elements, is used for dealing with collisions in a hash table?

- A. The hash table is empty

- B. The hash table is full and no new elements can be inserted
- C. The hash table is partially full and rehashing is needed before a new element is inserted
- D. Both (A) and (B) can never happen
- E. none of the above

_____ 14. Which of the following are true about open addressing collision resolution strategies?

- A. Linear resolution can produce severe clustering
- B. Quadratic resolution produces only moderate clustering
- C. Double hashing results in no clustering at all
- D. All three statements above are true
- E. none of the above statements is true

_____ 15. Suppose you find the following three nested loops in an algorithm:

```
for (int i = 1; i <= n; i++)
{
    for (int j = i; j <= n-1; j++)
    {
        for (int k = 1; k <= 10; k++)
        {
```

where n is an integer that is passed as an input to the algorithm. If all other statements in the algorithm are simple constant time operations, what is the likely complexity of the algorithm?

- A. $O(n)$
- B. $O(n^3)$
- C. $O(n \log n)$
- D. $O(1)$
- E. none of the above

_____ 16. Suppose you are asked to write a program to compute the function $f(x) = f(x-1) + f(x-2)$. Which of the following statements is true?

- A. An iterative program to compute this function will be slower than a recursive program that computes the same function
- B. A recursive program to compute this function will be slower than an iterative program that computes the same function
- C. An iterative program to compute this function will require quadratic time.
- D. A recursive program to compute this function will run in linear time.
- E. none of the above

_____ 17. Suppose x is an initially empty stack. Then suppose the following operations are done on x :

```
x.push(6); x.push(3); x.pop(); x.push(10); x.push(4);
x.push(7);

x.push(5); x.pop(); x.push(6); x.push(11);
```

Which of the following show the correct contents of the stack after all the operations have completed? (Assume the top of the stack is at the left.)

- A. 6, 10, 4, 7, 6, 11
- B. 6, 10, 4, 7, 11

- C. 11, 6, 6, 10, 4, 7
- D. 11, 6, 7, 4, 10, 6
- E. none of the above

Questions 18 through 20 refer to the following block of code:

```
public char[] mystery (char[] s, int i, int j)
// i is the first index of s
// j is the last index of s
{
    char c;
    if (j-i < 1)
        return s;
    else
    {
        c = s[i];
        // call mystery with a sub-array i+1..j of s
        mystery(s, i+1, j);
        for (int k = i+1; k <= j; k++)
            s[k-1] = s[k];
        s[j] = c;
    }
    return s;
}
```

_____ 18. What does this algorithm do?

- A. Reverses the order of characters in the array
- B. Flips the first and last characters of the array while leaving the other characters where they are
- C. Sorts the characters in the alphabetical order
- D. Creates a random permutation of the characters in the input array
- E. none of the above

_____ 19. If the algorithm is called with [p, q, r, s, t] as the input array of characters, what will be the array “slice” passed as parameter to the **first** recursive call to mystery?

- A. [p, q, r, s]
- B. [q, r, s, t]
- C. [p, q, r, s, t]
- D. [q, r, s]
- E. none of the above

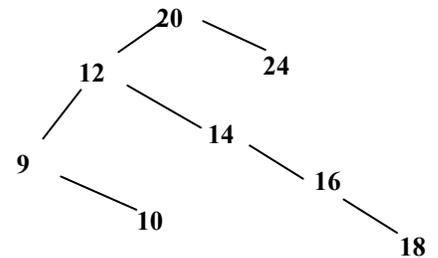
_____ 20. What is the order of complexity of this algorithm?

- A. $O(n \log n)$
- B. $O(1)$
- C. $O(\log n)$
- D. $O(2^n)$
- E. none of the above

Part II. Short Answer Questions

21. (2 pts) Suppose an algorithm requires $O(n^3)$ time on a computer. Will its time complexity order change if it is executed on another computer that runs twice as fast? Answer **yes or no** and then **explain** your answer.

22. (6 pts) For the binary tree shown below, list the order in which its nodes will be visited if:



- (1) A preorder traversal of the tree is done: _____

- (2) An inorder traversal of the tree is done: _____

- (3) A postorder traversal of the tree is done: _____

23. (4 pts) Write a recursive Java method to calculate the sum of the first n integers $1+2+3+\dots+n$ where n is greater than or equal to 1.

24. (8 pts) Describe, using a combination of Java and English, how insert (enqueue) and delete (dequeue) methods can be implemented to run in **constant time** ($O(1)$) for a **circular** queue implemented using an array.

25. (4 pts) If the array of integers [4, 3, 1, 5, 2, 0] is passed to selection sort, show how the array will look like after the **second** pass (second execution of the loop) of selection sort is completed:

```
public <T extends Comparable> void MysterySort (T[] data)
{
    int position, scan;
    T temp;

    for (position = data.length - 1; position >= 0; position--)
    {
        for (scan = 0; scan <= position - 1; scan++)
        {
            if (data[scan].compareTo(data[scan+1]) > 0)
            {
                temp = data[scan];
                data[scan] = data[scan + 1];
                data[scan + 1] = temp;
            }
        }
    }
}
```

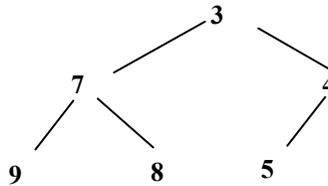
26. (3 pts) Which sorting algorithm is implemented by the method above?

27. (3 points) Suppose X is initially an empty queue. Then suppose the following operations are done on X:

```
x.enqueue(new Integer(6));  
y.enqueue(new Integer(3));  
Object y = x.dequeue();  
x.enqueue(new Integer(10));  
x.enqueue(new Integer(4));  
x.enqueue(new Integer(7));  
x.enqueue(new Integer(5));  
Object y = x.dequeue();  
x.enqueue(new Integer(6));  
x.enqueue(new Integer(11));
```

After these operations are completed show the contents of the resulting queue X:

28. (3 pts) If the number 2 is inserted into the min heap shown below, redraw the heap after the insertion and reordering are completed.



29. (3 pts) If the number 2 is inserted into a minimum priority queue that already contains the elements 8, 5, 9, 3, 7 and 4, a delete operation is done, then the number 6 is inserted and another delete is done, then what will be the contents of the priority queue after this last delete completes?

30. (4 pts) Suppose you need to find whether an array of integers that is already sorted in the ascending order and which contains no duplicates has any integer such that $A[i]=i$, that is an integer in a cell being the same as the index of that cell. You can find this out by making a pass through the entire array checking if $A[i]=i$ for all i . But you can do this more efficiently based on the property that the integers in the array are already sorted in the increasing order and that all integers are different. Can you come up with a faster and more efficient algorithm than iterating through the entire array? Please write your algorithm in English or in Java. Use the next page for more space

APPENDIX C

CS3 PRE TEST

Name: _____

Have you audited or taken COMP 3270 for credit before? (circle one) Yes No

**COMP 3270 — Introduction to Algorithms
Spring Semester 2009
Course Pretest**

The following pretest will help us assess your current knowledge of computing concepts and skills that will be covered in this course, so that we can better tailor this course to your needs. **You have 45 minutes to complete this test.** If you finish early, please bring your test to the proctor. For multiple choice questions, **write the letter (A,B,C,D or E) of the correct answer on the space to the left** of each question.

Important:

The purpose of this test is to identify the knowledge that you already have so that we can better tailor this course in future. Do not worry if you are unable to answer all of the questions. Answer a question if you know how to, or know enough to make a reasonable attempt. Otherwise leave it blank.

Do not guess blindly or choose randomly if you know nothing about a particular question.

Questions differ in their difficulty. Instead of answering them in the given order, do the easy ones first.

Your grade on this test will not be counted toward your course grade; however, you are required to complete this pretest as part of the course.

Please begin this test only when you are instructed to do so.

Use this page for scratch work

E. none of the above

_____ 6. Suppose a growth function of an algorithm is $T(n) = 1000\log n + 100\sqrt{n} + 3n^2 + n^2\log n + 5$. What is its asymptotic complexity or order in the Big-Oh notation?

- A. $O(\sqrt{n})$
- B. $O(n^2)$
- C. $O(\log n)$
- D. $O(n^2\log n)$
- E. none of the above

_____ 7. If a program reads in a sequence of integers, constructs a max heap from them, and then repeats the following three steps — (1) delete the root of the heap, (2) reorder the remaining heap, (3) then place the data element from the deleted root in the last cell that opened up at the end of the array containing the heap due to the removal of the root node from the heap — again and again until the heap consists of just one item, what is the order in which the data elements of the heap will now be arranged in the array?

- A. From the smallest to the largest
- B. In the reverse order as they were inputted into this program
- C. In the same order as they were inputted into this program
- D. From the largest to the smallest
- E. none of the above

_____ 8. Why does Radix Sort need a stable sorting algorithm?:

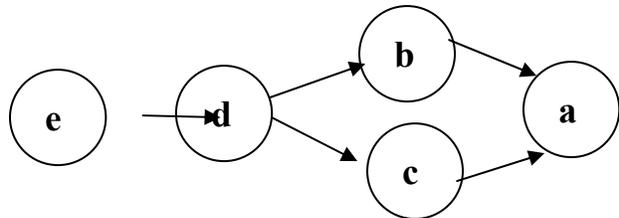
- A. A stable sorting algorithm ensures that the time needed by Radix Sort is stable
- B. A stable sorting algorithm ensures that the time needed by Radix Sort is independent of input size
- C. A stable sorting algorithm ensures that the space needed by Radix Sort is stable
- D. A stable sorting algorithm ensures that the space needed by Radix Sort is independent of input size
- E. none of the above

_____ 9. Which of the following statements about graph representation is **false**?

- A. An adjacency matrix representation of an undirected graph is symmetric, i.e., $A[i,j]=A[j,i]$
- B. If node a has an edge to node b in an undirected graph, its adjacency list representation will have node a in node b's linked list and vice versa
- C. If there is only one edge between node a and node b in a directed graph, its adjacency list representation will have node a in node b's linked list or vice versa but not both
- D. An adjacency matrix representation of a directed graph will never have $A[i,j]=A[j,i]$ for any i and j
- E. none of the above

_____ 10. What is the topological sort of this graph?

- A. a, b, c, d, e
- B. a, c, b, d, e
- C. e, d, a, b, c
- D. e, d, a, c, b
- E. none of the above



_____ 11. What is true of the input array of quick sort after the partitioning step?

- A. The part of the array to the left of the pivot are sorted

- B. The part of the array to the right of the pivot are sorted
- C. The part of the array to the left of the pivot contain elements greater than or equal to the pivot
- D. The part of the array to the right of the pivot contain elements less than or equal to the pivot
- E. none of the above

- _____ 12. Where will you find the maximum element on a binary search tree?
- A. The leftmost leaf
 - B. The rightmost node with no children or a left child
 - C. The root
 - D. The rightmost leaf
 - E. none of the above

- _____ 13. For a problem to be solvable by dynamic programming, it should have optimal substructure. What does this mean?
- A. It means that the problem can be divided into subproblems by making a choice
 - B. It means that there exists an optimal solution to the problem
 - C. It means that the optimal solution to the problem must include optimal solutions to the subproblems
 - D. It means all of the above
 - E. It means none of the above
- _____ 14. Why is it important for a problem to have overlapping subproblems in order for a dynamic programming algorithmic solution to be more efficient than a recursive solution?
- A. Overlapping subproblems suggest that a recursive algorithm will call itself many times with the same input, while the table lookup of dynamic programming avoids this
 - B. Overlapping subproblems suggest that a recursive algorithm will produce more than one solution, while the table lookup of dynamic programming will produce only one solution
 - C. Overlapping subproblems suggest that a recursive algorithm will not halt, while the table lookup of dynamic programming will halt
 - D. Overlapping subproblems suggest that a recursive algorithm cannot solve the problem because the subproblems are not independent, while the table lookup of dynamic programming can
 - E. none of the above
- _____ 15. When is a greedy algorithm more appropriate for a problem than a dynamic programming solution?
- A. When making a choice leaves no subproblems to be solved
 - B. A greedy algorithm is never more appropriate than dynamic programming
 - C. When making a choice leaves exactly two subproblems to be solved
 - D. When making a choice leaves more than two subproblems to be solved
 - E. none of the above
- _____ 16. If four algorithms to solve the same problem have the following complexities: $O(n^3)$, $\Omega(n^3)$, $o(n^3)$, $\Theta(n^3)$, which one is likely to be the most efficient?
- A. $O(n^3)$ algorithm
 - B. $\Omega(n^3)$ algorithm
 - C. $o(n^3)$ algorithm
 - D. $\Theta(n^3)$ algorithm
 - E. all of the above four algorithms are equally efficient
- _____ 17. Suppose you are asked to write a program to compute the function $f(x) = f(x-1) + f(x-2) + f(x-3)$; $x \geq 0$; $f(0) = f(1) = f(2) = 1$. Which of the following statements is **most** true?
- A. One can write either a recursive or an iterative (looping) program to compute this function
 - B. One can only write an iterative program to compute this function
 - C. One can write either a recursive or an iterative program to compute this function but the iterative program will be faster
 - D. One can only write a recursive program to compute this function
 - E. none of the above

Questions 18 through 20 refer to the following block of pseudocode:

```
Function Mystery(S[i..j]): an array of characters with array indexes
i..j)
C: character
begin
  if size(S) <= 1 then
    return S
  else
    C = S[i]
    Mystery(S[i+1..j])
  {comment: Mystery is called with a sub-array i+1..j of its input}
  for k=i+1..j do
    S[k-1] = S[k]
  end for
  S[j] = C
end if
return S
end Mystery
```

_____ 18. What does this algorithm do?

- A. Sorts the characters in the alphabetical order
- B. Flips the first and last characters of the array while leaving the other characters where they are
- C. Reverses the order of characters in the array
- D. Creates a random permutation of the characters in the input array
- E. none of the above

_____ 19. What are the **most accurate** recurrence relations of this algorithm (c_1 , c_2 , and c_3 are constants)?

- A. $T(n) = T(n+1) + c_1(n-1) + c_2$; $T(n \leq 1) = c_3$
- B. $T(n) = T(n-1) + c_1(n) + c_2$; $T(n \leq 1) = c_3$
- C. $T(n) = T(n-1) + c_1(n+1) + c_2$; $T(n \leq 1) = c_3$
- D. $T(n) = T(n-1) + c_1(n-1) + c_2$; $T(0) = c_3$
- E. none of the above

_____ 20. What is the order of complexity of this algorithm?

- A. $O(n \log n)$
- B. $O(1)$
- C. $O(n^2)$
- D. $O(2^n)$
- E. none of the above

Part Ib. Short Answer Questions

21. (4 pts) If a recursive algorithm has the following recurrence relations — $T(n) = T(n-1) + 1$; $T(0) = 1$ — state the polynomial $T(n)$ that characterizes its time complexity by solving the recurrence relations. Hint: forward or backward substitution — show at least one step.

22. (8 pts) Assuming union-by-size and path compression are used, illustrate the result of the following program by drawing the tree or trees that will result (MAKE-SET makes a disjoint set containing its argument; For UNION, if two trees have the same size make the second tree a child of the root of the first tree):

```
for i = 1 to 8
    MAKE-SET(i)
i = 1
repeat
    UNION(i, i+1)
    i = i+2
until i > 8
UNION(1, 3)
FIND(4)
```

23. (2 pts) Write the **exact** number of times the statement marked with a * in the following pseudocode fragment will be executed (n is a positive integer > 1):

```
i = n
repeat
*   i = i-1
until i = 0
```

24. (2 pts) Write the **exact** number of times the statement marked with a * in the following pseudocode fragment will be executed (n is a positive integer > 1):

```
i = 1
loop
    some statements...
    i = i+1
    if i < n then
*       go to loop
    end if
```

25. (2 pts) Write the **exact** number of times the statement marked with a * in the following pseudocode fragment will be executed (n is a positive integer > 1):

```
i = 1
```

```

*while i <= n
    i = i+1
end while

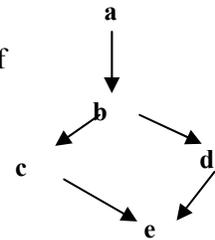
```

26. (6 pts) A long weekend is coming and you have the following activities for that Sunday.

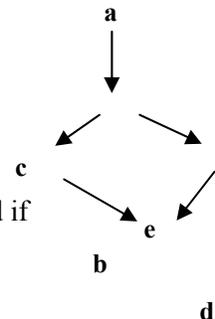
	Activity	Start time	Finish time
1	Swimming lesson I	7am	8am
2	Swimming lesson II	5pm	6pm
3	Grocery shopping	10am	11am
4	Window shopping at mall	3pm	5:30 pm
5	Picnic	9am	3:30 pm
6	Sun tan in front of apartment	3:30 pm	5:30pm
7	Ballgame at park	1pm	4pm

Assume all activities have the same priority and you would like to maximize the number of the activities you participate in. Additionally, assume the time required for commute to and between activities can be ignored. At most how many activities can you participate in, without schedule conflict? List these compatible activities.

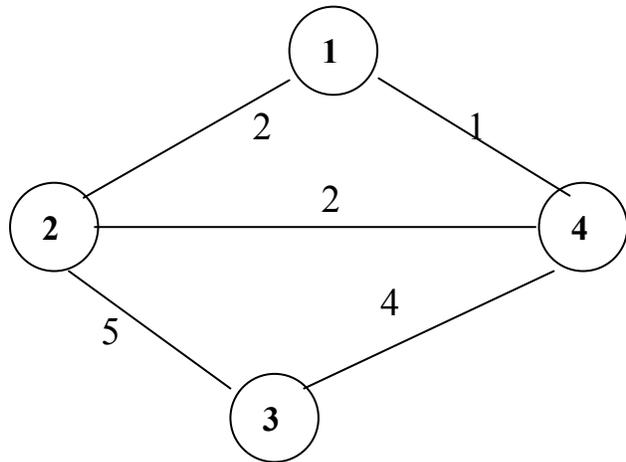
27. (4 pts) If depth first search is done on this graph starting with node a, and if adjacent nodes are visited in alphabetical order, draw the corresponding depth first search tree.



28. (4 pts) If breadth first search is done on this graph starting with node a, and if adjacent nodes are visited in alphabetical order, draw the corresponding breadth first search tree.



29. (2 points) Draw the minimum spanning tree of this graph.



30. (6 pts) Suppose you need to find whether an array A of non-negative integers that is already sorted in the ascending order, and which contains no duplicates, has any integer such that $A[i]=i$, that is an integer in a cell being the same as the index of that cell. You can find this out by making a pass through the entire array checking if $A[i]=i$ for all i . But you can do this more efficiently based on the property that the integers in the array are already sorted in the increasing order and that all integers are different. Can you come up with a faster and more efficient algorithm than the obvious one of iterating through the array until you find an index i such that $i=A[i]$ or keep going till you reach the end? Write your algorithm in English or pseudocode, not in a programming language.