PROBLEM-BASED LEARNING INSTRUCTION VERSUS TRADITIONAL INSTRUCTION ON SELF-DIRECTED LEARNING, MOTIVATION, AND GRADES OF UNDERGRADUATE COMPUTER SCIENCE STUDENTS

by

Noel F. LeJeune

B. S., Louisiana State University, 1970

B.S., Metropolitan State College, 1988

MCIS, University of Denver, 1993

A thesis submitted to the

University of Colorado at Denver

in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Educational Leadership and Innovation

2002

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

This thesis for the Doctor of Philosophy

degree by

Noel F. LeJeune

has been approved

by

Ellen Stevens

Ellen Stevens

Laura D Goodwin

Laura Goodwin

James T. Loats

James T. Loats

Michael Marlow

Michael Marlow

April 19, 2002

Date

LeJeune, Noel F. (Ph.D., Educational Leadership and Innovation)

Problem-Based Learning Instruction Versus Traditional Instruction on Self-Directed Learning, Motivation, and Grades of Undergraduate Computer Science Students

Thesis directed by Associate Professor Ellen Stevens

ABSTRACT

A problem-based learning (PBL) teaching method was compared with a traditional lecture-based teaching method to determine the effects on undergraduate Computer Science (CS) students' self-directed learning (SDL) and programming assignment grades. An integrated construct of SDL included a) SDL readiness b) SDL skills, c) SDL performance, and d) students' course motivation.

Quasi-experimental designs were used to compare a PBL teaching method and a traditional lecture-based method in two sections of a CS1 course taught by the same instructor. Each of the SDL components and grades were measured for students experiencing traditional instructional methods and problem-based learning methods. Readiness was measured with the Self-Directed Learning Readiness Scale, skills with the Motivated Strategies for Learning Questionnaire–Part B, performance with time spent on SDL tasks, and course motivation with the Motivated Strategies for Learning Questionnaire–Part A. The grade measurement was the course instructor's percentage score given to students' programming assignments.

Results showed that the effect of teaching method was statistically significant for the SDL performance measure with the PBL section demonstrating greater performance. The effect of teaching method was not significant on SDL readiness, skills or course motivation measures. A lack of statistical differences between the two methods for these measures was attributed to no effect of PBL on students' SDL or small sample size resulting in reduced statistical power or lack of student engagement in PBL resulting in ineffective treatment.

The effects of method, time, and method x time interaction were significant on the grades measure. The traditional teaching method group had higher grades than the PBL group. Both teaching methods exhibited declining grades over time. Factors such as increased difficulty of assignments and stricter grading schemes over time or differing characteristics of group members such as prior CS knowledge, age, time spent on assignments, and competing employment and other course demands were identified.

Recommended future study includes improved measures of students' SDL practices rather than students' own perceptions, assessment of student practice of PBL, and qualitative study of students' motivation and SDL performance.

This abstract accurately represents the content of the candidate's thesis. I recommend its publication.

Signed Ellen Stevens

Ellen Stevens

DEDICATION

I dedicate this dissertation to my wife. Lynn Callaway. who. having completed her dissertation. never let me forget the definition of a good dissertation.

ACKNOWLEDGEMENT

The faculty of the Graduate School of Education at the University of Colorado at Denver has provided wonderful support and inspiration in my studies. Particular thanks goes to my dissertation committee members, Laura Goodwin, Jim Loats, and Mike Marlow. Special appreciation goes to my advisor, Ellen Stevens, for her guidance, support, wisdom, and knowledge so freely shared throughout my entire doctoral program. The "PSTL" Lab partners have been a valued source of discourse and support.

Dr. Patricia Tucker, in addition to providing encouragement and support, allowed me to "experiment" with her classes and accommodated many requests that made the research possible. Professor Ruth Yara has also been a strong supporter and source of inspiration for many years. Dr. Charlotte Murphy, Chair of the Department of Mathematical and Computer Sciences at MSCD, is most appreciated for her support and FRIP sponsorship in completing this work while I served on the faculty at MSCD. My colleagues in the Department of Mathematical and Computer Sciences at MSCD have also kept the spirit alive. Dr. Shahar Boneh, Statistician in the MSCD Mathematical and Computer Sciences Department, also provided excellent consultation on statistical analyses. Finally, thanks to Dr. Tucker's students for their participation.

CONTENTS

APPENDIX

FIGURES

TABLES

CHAPTER 1

INTRODUCTION

Purpose

Computer Science graduates need strongly developed problem-solving skills, collaboration skills, and self-directed learning abilities to be successful professionals in the workforce (Hartman & White, 1990; Shaw, 2000). Once on the job, software developers need to continually update their knowledge to remain competent in a world of rapid growth and change. Since constant formal education by itself is often impractical for these continual learning challenges, graduates who also develop self-directed learning skills will be better prepared to respond to change than those who do not (Pomberg, 1993).

Undergraduate Computer Science education focuses on the technical aspects but may inadequately prepare students for continued self-directed learning (Shaw, 2000). The typical undergraduate curriculum includes coursework covering such topics as programming, algorithms, data structures, software design, concepts of programming languages, computer organization, and computer architecture (Computing Science Accreditation Board, 2001). Academia and industry (Shaw, 1976) recognize deficiencies in communication skills, collaborative skills and problem-solving skills (Hartman & White, 1990; National Science Foundation Advisory Committee, 1998). Undergraduates must experience these "soft" skills as well as the technical or "hard" skills (Wilson, Hoskin, & Nosek, 1993).

Providing these additional skills is necessary if Computer Science graduates are to be successful and competent in this ever-changing profession. Hartman (1990) suggested, "... the two most important skills which a [Computer Science] student embarking on his career can have are communication skills and problem solving skills. Without either of these things he is doomed to failure, or, at best mediocrity" (p. 216). Shaw (2000) added, "... traditional [Computer Science] education makes scant provision for helping students keep their knowledge current" and that "... pressures on educational institutions will require changes in what we teach software developers and how we teach it" (p. 373). Therefore, helping students become better self-directed learners must be a priority for Computer Science educators.

Traditional teacher-centered instruction using lecture and outside of class programming assignments do little to foster soft skills development. Problem-based learning, on the other hand, as a teaching technique could be a natural extension to many existing Computer Science courses where programming assignments are the norm. Problem-based learning provides opportunities for both collaborative learning and the development of problem-solving skills

(Cockrell, Hughes Caplow, & Donaldson, 2000; Jonassen & Kwon, 2001; Koschmann, Kelson, Feltovich, & Barrows, 1996). There are claims that problem-based learning also develops self-directed learners (Barrows, 1994). While this relationship seems logical, specific research on the influence of problem-based learning teaching techniques on students' self-directed learning is needed.

Students taught with problem-based learning are better able to apply their knowledge (Schmidt, 1983), are better problem solvers (Albion & Gibson, 1998; Vernon & Blake, 1993), and develop better communication skills (Lieux, 1996). However, a question remains concerning self-directed learning: What is the influence of problem-based learning on students' self-directed learning? This question suggests the overarching topic for this research. Furthermore, this work focuses specifically on the use of problem-based learning teaching methods for undergraduate Computer Science education. The conceptual framework in the following section situates this study within the context of developing better self-directed learners while using problem-based learning. (Detailed discussions of problem-based learning and self-directed learning are found in chapter 2.)

Conceptual Framework

The conceptual framework provides a context for the research and the basis for each research question. The framework describes self-directed learning as an integrated construct located both within and separate from problem-based learning.

Educating prospective computer scientists requires a more holistic approach than merely teaching the principles and practices of the profession. The failures and shortcomings of many graduates often result from poorly developed "soft" skills rather than from deficiencies in the principles of Computer Science (Hartman & White, 1990). The more notable soft skills students need are problem solving, collaborative skills, well-developed communications skills, and self-direction (Shaw, 2000).

A new approach to Computer Science education should provide both the domain knowledge as well as opportunities to learn and develop these soft skills. With this goal comes the question: "How can we teach the core body of knowledge while, at the same time, developing better self-directed learners?" As Shaw (1976) suggested, one possible solution involves our teaching methods. We should employ methods that give students opportunities to develop their self-directed learning, develop their communication skills, become more accomplished problem solvers, and learn the subject matter. Problem-based learning teaching methods may offer specific solutions to this conundrum.

Problem-Based Learning

Problem-based learning is defined as an instructional technique that uses ill-defined, complex problems as the impetus for learning (Barrows, 1994; Koschmann et al., 1996; Ram, 1999). Students collaboratively define learning issues, define and use learning resources, and share acquired knowledge with the guidance of a tutor/facilitator. Students create solutions for the problem. Plenary sessions foster review and reflection upon the learning as well as the problem-based learning process itself. Problem-based learning provides experience in problem solving, collaborative work, self-direction, and teaches students subject matter content.

Problem-based learning methods develop problem-solving skills while they also teach students the subject matter. Furthermore, problem-based learning uses collaborative learning, thus providing valuable experience with another critical soft skill. Problem-based learning also uses specific self-directed learning skills noted as the "skills" component of self-directed learning in Figure 1.1. Skills such as problem recognition and learning resource identification and acquisition that are used during problem-based learning (Barrows, 1994) are also used by self-directed learners (Rutland & Guglielmino, 1987). These additional aspects are illustrated in Figure 1.1.

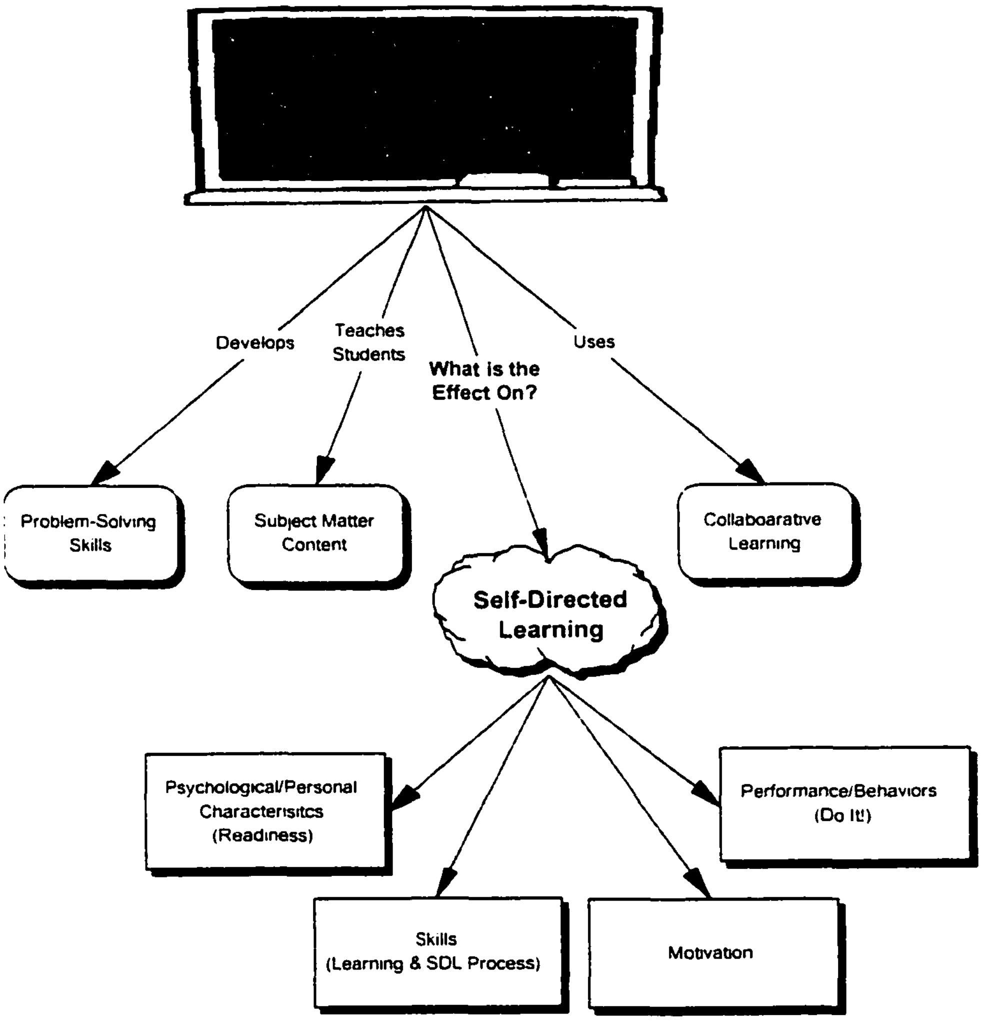

Figure 1.1 Problem-Based Learning Teaching Strategy

The nature of the activities in problem-based learning suggests that the use of problem-based learning teaching methods should have a positive influence on the development of self-directed learning skills. Other facets of self-directed learning such as readiness, actual performance, and motivation may also be affected by the use of problem-based learning.

Self-Directed Learning

Self-directed learning has many descriptions. It is recognized as an instructional method (Knowles, 1975), a personality attribute of the learner (Brockett & Hiemstra, 1991; Candy, 1991; Hiemstra, 1992), or a process for learning (Knowles, 1975). Knowles defined self-directed learning as "a process in which individuals take the initiative, with or without the help of others, to diagnose their learning needs, formulate learning goals, identify resources for learning, select and implement learning strategies, and evaluate learning outcomes" (1975, pg.18). Candy (1991) combines both the personal attributes of personal autonomy and self-management with the learning activities of self-instruction and learner-control to describe self-directed learning. Grow (1991b) defines self-directed learners within an institutional setting as "those who, within a teacher-controlled setting, take greater charge of their own motivation, goal-setting, learning, and evaluation" (p. 203). Any comprehensive definition should recognize both the personal attribute and the instructional method since they are related and inseparable (Pilling-Cormick, 1996).

Synthesizing these views reveals self-directed learning as a combination of capabilities and motivations of the learner. An integrated definition of self-directed learning is key to understanding what is necessary to be a self-directed learner. Any such view must include the following four components: 1) traits of the learner that reflect on his or her propensity toward self-directedness (or readiness), 2) capabilities or skills of the learner to undertake a self-directed project, 3) the actual performance (behaviors) of the process of undertaking the self-directed learning project, and 4) the individual's motivation toward the learning project. Figure 1.2 illustrates the integrated construct for self-directed learning. The psychological/personal component is the individual's readiness for self-directed learning. The skills include both basic learning skills and those necessary for conducting a self-directed learning activity such as defining learning goals, finding the necessary resources, conducting the learning activities, and self-assessing the process and learning. Performance/behavior is the component that represents the actual doing self-directed learning. A propensity toward self-directed learning (readiness) and having the skills to conduct self-directed learning do not make one a self-directed learner. However, it is the actual doing it that makes one a self-directed learner. This is the performance and behaviors demonstrated by a self-directed learner. Putting these potential attributes in practice also requires motivation.

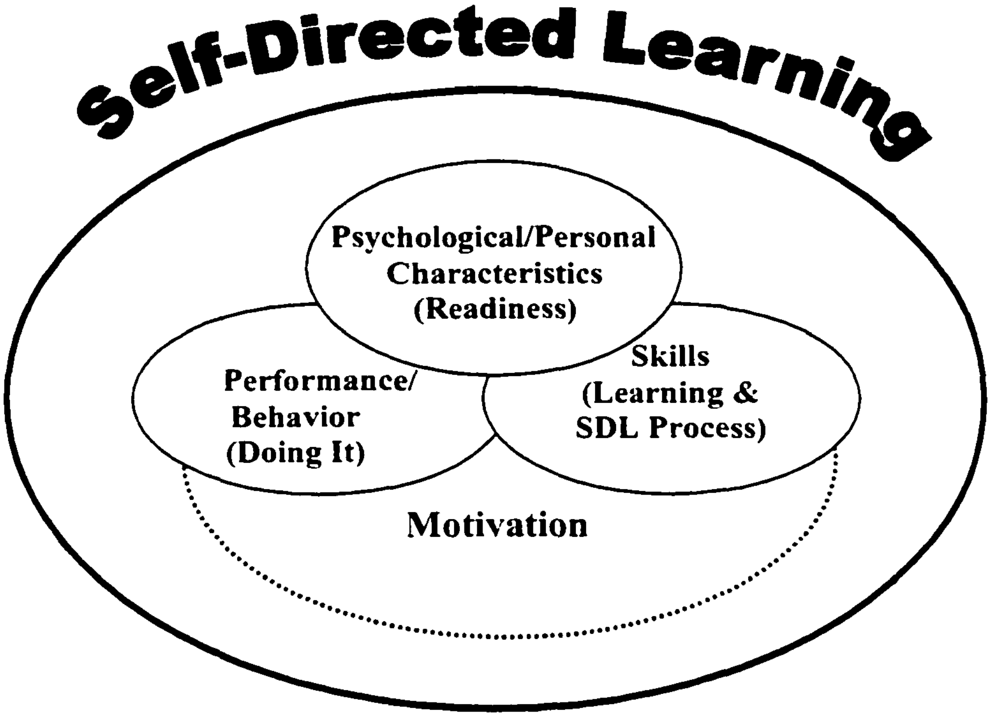
Self-Directed Learning
Psychological/Personal Characteristics (Readiness)
Performance/ Behavior (Doing It)
Skills (Learning & SDL Process)
Motivation

Figure 1.2. Components of Self-Directed Learning

Motivation plays a significant role in the practice of self-directed learning (Long, 2001; Pintrich, 1995). While motivation is portrayed as a discrete component within the overall concept of self-directed learning in Figure 1.2, it overlaps each of the other components since it affects all. The dotted outline suggests it permeates the entire construct. Motivation influences an individual's perception of their skills to accomplish the task at hand. Performance is also strongly affected by the perception of skills as well as the actual skills of the learner (Bitterman, 1988; Confessore, 1991; Long, 2001). Self-efficacy is a key element in one's actual performance (Hoban, Sersland, & Raine, 2001). Since "... people rarely choose to do tasks that they expect to fail" (Stipek, 1998, p. 137), self-directed learning performance is tightly linked to motivation. Motivation also varies with the context of the activity and the learner's perceived needs (Long, 2001).

This proposed concept for a multi-dimensional construct includes the potential for self-directed learning such as Guglielmino's Readiness as a psychological or personal characteristic, the skills (actual or self-perceived), the driving motivation factor, and the actual performance.

Summary

Computer Science education needs teaching strategies that provide the opportunity to develop students' self-directed learning while teaching the subject matter content. A teaching strategy using problem-based learning is feasible for many Computer Science courses, especially the many courses that require programming assignments. With a problem-based learning teaching method, it may be possible to better develop students' self-directed learning while teaching the subject matter. An integrated, multi-dimensional view of self-directed learning includes components of personal characteristics, skills, behavior, and motivation. The connection between problem-based learning and the components of self-directed learning for undergraduate Computer Science students has yet to be examined. This issue constitutes the main topic of inquiry in this work. The specific research questions and hypotheses are stated in the next section.

Research Questions

The overarching research question for this study is: "What are the changes in undergraduate Computer Science students' self-directed learning characteristics after experiencing problem-based learning?" With this question and the conceptual framework above, specific research questions are:

1. Are there significant differences between students experiencing a problem-based learning teaching method and students experiencing traditional lecture-based teaching method on:
    a. Students' self-directed learning readiness?
    b. Students' self-directed learning skills?
    c. Students' self-directed learning performance?
    d. Students' course motivation?
    e. Students' programming assignment grades?
2. Are there significant differences across time on:
    a. Students' self-directed learning readiness?

b. Students' self-directed learning skills?

c. Students' self-directed learning performance?

d. Students' course motivation?

e. Students' programming assignment grades?

3. Is there a significant interaction between teaching method and time on:

a. Students' self-directed learning readiness?

b. Students' self-directed learning skills?

c. Students' self-directed learning performance?

d. Students' course motivation?

e. Students' programming assignment grades?

Methodology

A quasi-experimental design was used to determine the differences in self-directed learning, course motivation, and grades of undergraduate Computer Science experiencing problem-based learning versus students experiencing traditional teaching methods. The design compared teaching methods of problem-based learning method (the treatment) with a traditional lecture-based teaching method for the control group.

Two regular programming assignments for the course were modified to create two successive problem-based learning assignments; this and the traditional lecture method constitute the independent variable. Five dependent variables, each of which originates from the research questions are self-directed learning readiness, self-directed learning skills, self-directed learning performance, motivation, and grades. The independent and dependent variables are fully defined in Chapter 3, Methodology.

Structure

Chapter 1 has presented an overview of the purpose of the study, background information suggesting the need for the study, a conceptual framework, the research questions, operational definitions, and an overview of the methodology. Chapter 2 provides a review of the pertinent literature. Chapter 3 describes the methodology including the design, instruments for measurement, experimental procedures, and methods for analysis. Chapter 4 contains the findings. Chapter 5 summarizes the findings and presents the answers to the research questions. This final chapter also discusses the implications of these results for future practice and future research.

CHAPTER 2

LITERATURE REVIEW

This chapter begins by examining self-directed learning and the components included in an integrated view of self-directed learning. The discussion then moves to problem-based learning as a teaching method. Next, the "Relationships" section describes how self-directed learning components are related to problem-based learning activities to suggest an expectation of positive changes in students' self-directed learning after a problem-base learning experience as compared to the traditional teaching method.

Evidence suggests problem-based learning experiences might positively influence students' self-directed learning (Barrows, 1994; Blumberg & Michael, 1992; Ryan, 1993; Taylor, 1986) in part because problem-based learning activities share some skills and behaviors with those of self-directed learning (Barrows, 1994; Hmelo, Gotterer, & Bransford, 1997). For example, the goal orientation found in descriptions of problem-based learning seems similar to motivation as described for self-directed learning.

Self-Directed Learning

The study of self-directed learning has a rich and continually evolving history in spite of a lack of a consensual definition (Bulik & Romero, 2001; Long, 1990). Most authors credit Houle's work reported in 1960 as the beginning of modern-day investigations (Confessore & Confessore, 1992a, 1992b). Houle (1988) proposed self-directed learners do so either to satisfy a goal, for the sake of learning itself, or for the enjoyment of the learning environment and activity. Tough (1978) followed with a seven-year study of frequencies and methods of self-directed learning projects. Self-directed learning has been examined from viewpoints ranging from psychological traits (Brockett, 1985; Brockett & Hiemstra, 1991; Candy, 1991; Hiemstra, 1992) to instructional methods (Knowles, 1975) to a teaching—learning situational construct (Pilling-Cormick, 1996). The complexity of self-directed learning has been acknowledged recently by Long (2001) who posits it may involve all of those viewpoints. Some of these definitions are briefly reviewed in the next section before discussing an integrated view self-directed learning.

Definitions of Self-Directed Learning

Knowles recognized self-directed learning as an instructional method when he described the processes for conducting a self-directed learning project (Knowles, 1975). He also linked his concept of adult learning (andragogy) to psychological traits he associated with adult learners, for example, a desire among adults for greater responsibility for their own learning (Knowles, 1990). Others with a greater emphasis on the psychological or personality perspective of self-directed learning include Guglielmino (1977), Brockett and Hiemstra (1991), and Candy (1991). However, Brockett and Hiemstra (1991) also recognized instructional methods and learner traits as important. Even their Personal Responsibility Model (PRO) emphasizing psychological characteristics distinguished the teaching-learning transaction as "self-directed learning" while the primary characteristics of the student were labeled "learner self-direction". Candy (1991) also combined the personal attributes and the learning activities; personal autonomy in the form of willingness and self-management was the primary focus for self-directed learning. Candy, on the other hand, restricts the learning activities of self-directed learning to non-institutional settings where the learner had only self-imposed structure and requirements. Grow (1991b) is less restrictive and included institutional settings as valid environments for self-directed learning. Grow (1991b) included psychological, process, and environmental factors when defining self-directed learners as "those who, within a teacher-controlled setting, take greater charge of their own motivation, goal-setting, learning, and evaluation" (p. 203). Pilling-Cormick (1994; 1996) emphasized the environmental factors that were either conducive or detrimental to self-directed learning while recognizing both personal attributes and instructional methods (Pilling-Cormick. 1996). Her model for self-directed learning primarily focused on the process with three major components: educator, student and locus of control (student versus teacher). Cranton's (1992) description of self-directed learning emphasizes the process, outcomes, and goals.

With many different views of self-directed learning throughout the literature, it is difficult to arrive at a single definition. While the views seldom conflict on substantive issues, each expert has a viewpoint or specialty that may represent only a part of the complex whole. An integrated definition of this complex, multi-faceted concept follows.

Integrated Definition of Self-Directed Learning

A definition restricted to any one of the many traits associated with self-directed learning is inappropriate. Long (2001, p. 10) offered a restatement of his theoretical position concerning self-directed learning by recognizing the complexity of the topic. An integrated, possibly holistic, view emerges.

> Many variables may affect the manifestation of self-direction. They seem to include, but are not limited to, [italics added] (a) personality and other psychological constructs, (b) aptitude and familiarity with the content to be learned, (c) learning context including powerful others' expectations, teaching techniques employed, and degree of learner autonomy and control. Other variables are (d) social relation with other learners, facilitator, and other resource people, and (e) immediate personal and professional situation in which the learner finds himself or herself

Not only is the complexity recognized in his statement, but also the need to address other components is noted with the "not limited to" statement. The most commonly discussed aspect of self-directed learning has been the psychological/personal trait listed first in Long's list. A skills component is suggested within Long's items (b) and (d) while motivation appeared in item (c). He also stated "motivation may be more important than current research indicates by the few studies dealing with the topic" (p. 9).

The integrated construct of self-directed learning included a combination of a) psychological/personal traits for self-directedness, b) skills or capabilities for conducting one's own learning projects, c) performance/behaviors applying those skills to the self-directed learning activities, and d) motivation for the particular learning project. While there were likely other variables as Long suggests, these four were the most prominent and significant throughout the literature. The remainder of this section discusses each of these four components.

Personal/Psychological Characteristics. Guglielmino's (1977) definition focused on personal characteristics represented in the Self-Directed Learning Readiness Scale (SDLRS). The eight characteristics are:

1. Openness to learning opportunities
2. Self-concept as an effective learner
3. Initiative and independence in learning
4. Informed acceptance of responsibility for one's own learning

5. Love of learning
6. Creativity
7. Positive orientation to the future, and
8. Ability to use basic study skills and problem-solving skills

These eight factors incorporate two of the integrated characteristics—personality and skill; seven of the eight are related to personality. This leaves performance and motivation missing from Guglielmino's conception of self-directed learning. However, over 70% of the self-directed learning methodological research from the last two decades focused on the Self-Directed Learning Readiness Scale (Brockett et al., 2000).

Self-Directed-Learning Skills. Some set of skills is necessary to conduct a self-directed learning project whether as a completely independent project or within a formal institutional setting. The specific skills can be inferred from analyzing the process followed by self-directed learners. Knowles (1975) described five activities that represent the core skills necessary:

> The process in which individuals take the initiative with or without the help of others, in diagnosing learning needs, formulating learning goals, identifying human and material resources for learning, choosing and implementing learning strategies, and evaluating learning outcomes (p. 18).

First in practice is formulating learning goals. The learner must have the ability to determine the goals from the context of the situation. Often this implies defining the problem that must be solved. Once the problem is known, a self-directed learner must recognize the knowledge and skills that must be acquired to solve the problem. A comparison of what one currently knows and does not know constitutes diagnosing learning needs. Knowles referred to this activity as a "Gap Analysis" (Knowles, 1975). After diagnosing learning needs, self-directed learners must have the ability to identify human and material resources for learning. Two sets of skills are needed; those to identify resources and those to use the resources. The use of resources transitions into the actual skills for learning. This represents the tasks of choosing and implementing appropriate learning strategies. Learning strategies include such activities as rehearsal, elaboration, organization, critical thinking, and metacognitive self-regulation (Pintrich & DeGroot, 1990). Ancillary learning strategy skills required for the successful self-directed learner involve resource management, effort regulation, help seeking, peer learning, and time management (Pintrich, Smith, Garcia, & McKeachie, 1991).

Self-Directed-Learning Performance/Behavior. A self-directed learner not only has the readiness and skills for self-directed learning, he or she does it. Students can be guided through the activities and taught the skills of self-directed learning (Grow, 1991c; Rutland & Guglielmino, 1987). Candy (1991) asserted, "one learns responsibility and self-direction through experiences in which one is given the opportunity to be self-directed and responsible for one's actions" (p. 319). One commonly used tool for practicing self-directed learning with a well-structured process is the learning contract (Blackwood, 1994; Caffarella, 1983; Caffarella & Caffarella, 1986; Guglielmino & Guglielmino, 1994). Learning contracts make the process of self-directed learning explicit and visible. Grow's (1991b) model for teaching self-direction suggests several other techniques that emphasize matching the students' level of self-direction with corresponding teaching methods and classroom activities. This form of scaffolding keeps students within their zone of proximal development (Vygotsky, 1978) for optimizing learning, avoiding frustration, and positively contributing to student motivation (Pintrich & Schunk, 1996, pp. 74, 175).

Motivation

The motivations for self-directed learning were first described by Houle (1961). He suggested three reasons learners pursue self-directed learning projects: a) to satisfy a goal or need (goal-oriented), b) for the love of learning (learning-oriented), or c) for the experience and enjoyment of the learning activities and associated ambience (activity-oriented). Houle acknowledged these motivating reasons are not mutually exclusive so a learner may be moved to participate in self-directed learning projects by combinations of these. Of these, goal-orientation has been identified as a significant factor in self-directed learners pursuing degrees from higher education institutions (Grow, 1991b; Ponton, Carr, & Confessore, 2000).

Goal orientation was seen as the student's perception of reasons for engaging in the learning task. While it is only one of several components of motivation, it appeared especially significant to self-directed learners (Bitterman, 1988). Other internal motivational factors included the perceived value of the task as the learner's evaluation of how interesting, how important, and how useful the task itself is. Self-directed learners generally recognize a need and are able to perceive value in the tasks. Expectancy components of motivation included students' belief that their efforts would result in positive outcomes, that their performance expectations would be met, and that their self-efficacy for the task was sufficient (Pintrich et al., 1991). Extrinsic goal orientation factors include grades, rewards, performance assessment, and evaluations.

Summary of Self-Directed Learning

An integrated view of self-directed learning is preferable to one that examines a single aspect of the concept. However, the possible number of components and their relationships suggests a holistic viewpoint may be more accurate than a simple integrated one. Nevertheless, a reasonable reduction of complexity results in a model with four predominate variables standing out in the literature. This model for self-directed learning is comprised of psychological/personal traits, a set of skills, a recognizable performance/behavior, and motivation.

Problem-Based Learning

Problem-based learning originated with medical education at McMaster University in the mid-1960s with the intent of improving students' problem solving skills while teaching basic subject matter content (Caplow, Donaldson, Kardash, & Hosokawa, 1997). The fundamental precept was that learning proceeded from the "need to know" in order to solve a problem, thus enhancing learning. Charlin, Mann, and Hansen (1998) said that problem-based learning (a) requires active processing of information, (b) activates prior knowledge, (c) provides a meaningful context, and (d) stimulates opportunities for elaboration and organization of knowledge. In addition to these learning benefits, problem-based learning provided experiences in problem solving, opportunities for collaborative work, and use of communications skills.

Many other disciplines including engineering, education, and the sciences have experimented with or adopted problem-based learning methods as part of courses or entire programs (Allen, Duch, & Groh, 1996; Arambula-Greenfield, 1996; Cawley, 1997; Groh, 2000; Grundy, 1996; Todd, 1997; Woods, 1996).

Definition and Characteristics of Problem-Based Learning

Although there are many variations, Albanese and Mitchell (1993, p. 53) defined problem-based learning as "... an instructional method characterized by the use of patient problems as a context for students to learn problem-solving skills and acquire knowledge about the basic and clinical sciences". Problem-based learning has been distinguished from other problem-centered methods such as the case method, in that the problem provides the motivation for learning basic concepts. The problem is presented before the learner is exposed to the subject or content knowledge. The need to understand the problem drives learning.

As early as 1986, Barrows (1986)offered a taxonomy for problem-based learning recognizing "... the many variables possible can produce wide variations in quality and in the educational objectives that can be achieved" (p. 481). This taxonomy can "... help teachers choose a problem-based learning method most appropriate for their students" (p. 481). Table 2.1 illustrates the range of problem-based learning instruction and provides a basis for selecting characteristics to incorporate in problem-based learning. This taxonomy represents the broadest range ofwhat may be included in problem-based learning.

Table 2.1. Barrows Taxonomy of Problem-Based Learning.

Lecture-Based Cases	Teacher presents information in lectures plus a case or two (vignettes) to demonstrate relevance – Not usually considered problem-based learning.
Case-Based Lectures	Case vignettes or more complete case histories are presented before lecture. Students analyze existing knowledge prior to lectures of new material.
Case Method	Students are given a complete case for study and research in preparation for subsequent class discussion.
Modified Case-Based	Small tutorial groups of students are presented with a case. Students pursue limited lines of inquiry from alternatives presented. Additional information is provided as requested by students.
Problem-Based	Problems are presented within an authentic context. Students use free inquiry. Active, teacher-guided exploration and evaluation using facilitation and tutorial skills is used.
Closed-Loop, Problem-Based	Problem-based as above with iteration cycles where each cycle concludes with students reflecting and evaluating 1) resources used, 2) reasoning processes followed, and 3) information acquired (learning).

The problem. The problem is central to the concept of problem-based learning. Authentic problems in an authentic context are used to develop content knowledge, problem solving skills, collaborative skills, and learner self-direction (Barrows & Tamblyn, 1980). The problems, in addition to being crafted with the learning objectives in mind, are complex, and ill-structured (Barrows, 1994; Koschmann et al., 1996; Stepien & Pyke, 1997). These types of problems are not fully or clearly understood by the students at the outset. Learners must extract or define the problem from the body of information initially available. As more information is acquired, the problem definition is likely to evolve along with a better understanding of both the problem and the knowledge needed for resolution. Another problem characteristic is that there may not be a single, simple, and "correct" solution or the "correct" solution is not likely to be known except in hindsight. Finally, these types of problems are not likely to have a fixed or previously established procedure for reaching a solution. The problem-solvers must, at least, define an approach to the problem from very high level approaches and, at most, determine an entirely new approach.

Learning Objectives

Learning objectives must be incorporated within the problem. Defining learning objectives for a problem-based learning experience range from wholly student defined to wholly teacher defined (Blumberg, Michael, & Zeitz, 1990; Blumberg & others, 1990; Duek & Wilkerson, 1991). In any case, the problem is designed to incorporate learning objectives appropriate to the module, course or program (Dolmans, 1993; Dolmans, Snellen-Balendong, Wolfhagen, & van der Vleuten, 1997; Stepien & Pyke, 1997).

Tutors

Effective problem-based learning methods do not rely on students following the process without direction and support. Tutors provide guidance and direction by working closely with each small group during the problem identification, learning issues definition, and reflection activities. An important responsibility of the tutor is to emphasize the problem-based learning processes rather than teach subject matter content. The tutor's primary role must be guiding students through the use of metacognitive skills needed for the problem at hand and for future practice. "This concept of metacognitive thinking skills provides the key to the positive, active role of the tutor" (Barrows, 1988, p. 3).

Tutors must be skilled in both the problem-based learning method as well as in reasoning skills (Barrows, 1988). Other tutor tasks include the use of questions to promote reasoning and critical thinking skills, facilitation of the group processes without directing, and assuring that the groups' derived processes are externalized. Tutors should distinguish the role of a content knowledge expert and the problem-based learning tutor role. Writers in the field are divided on whether the tutor and the course instructor should be the same person (Dolmans & others, 1993; Dolmans & Others, 1994, 1996; Gijselaers, 1994; Moust & Schmidt, 1995; Schmidt, van der Arend, Kokx, & Boon, 1995; Schmidt & Moust, 1995; Schmidt & Others, 1993; Wilkerson, 1995; Wilkerson, 1996; Wilkerson & Hundert, 1997). However, students tutored by subject matter experts are somewhat better achievers and tend to spend more time on self-directed learning (Schmidt & Others, 1993). Problem-based learning requires the learners to seek out the content knowledge as part of the learning experience although they will call upon subject matter experts as a resource. Finally, a tutor should stimulate reflection in the group on the newly acquired knowledge, the relationship and integration of this new knowledge with previous knowledge, and the potential for application (Barrows, 1988). The group reflection also seeks to uncover new learning needs. Lastly, the tutor encourages the learners to assess their problem solving skills, their processes as problem-based learners, and discover areas for improvement (Barrows, 1994). Problem-based learning studies indicate that this reflection component is necessary for successful learning outcomes (Barrows, 1994, p. 74).

The Process

The problem-based learning process begins when an authentic problem is presented to a small group of students. The group size is generally four to seven students, however variations for larger groups have been reported (Rangachari, 1996; Woods, 1996). Once students are presented with the problem, they perform an analysis to determine what they collectively know about the problem and what they need to know to solve the problem. This phase requires extensive collaboration and communication within the group. The group's efforts to define their existing knowledge and the knowledge needed to solve the problem provide experience with self-directed learning skills. Next students individually utilize resources they discover for themselves and with the assistance of the problem-based learning facilitation to acquire the knowledge and skills necessary to solve the problem. The group members then reconvene to share their individually acquired knowledge and continue the problem solving activity, again using collaborative learning. This cycle of assess—acquire—share repeats until a satisfactory solution is achieved. A key element is a reflection activity that concludes the problem-based learning process. This last stage, critically necessary for learning, consists of self and peer evaluation of abilities as problem-solvers, self-direction, and as members of the group (Barrows, 1994).

Benefits and Drawbacks of Problem-Based Learning

The most commonly cited benefit of problem-based learning is an increased ability to apply the knowledge acquired using problem-based learning (Albanese & Mitchell, 1993; Barrows, 1994; Hmelo et al., 1997). Also, the development of reasoning skills in problem solving is coupled with the ability to use knowledge in practice (Dolmans et al., 1997). Problem-based learning students demonstrated a higher hypothesis-driven reasoning ability than data-driven reasoning (Hmelo et al., 1997). This ability to work with an initially limited set of data to formulate both problem and possible solutions represents real-world situations better than is possible in traditional, lecture-based instruction. With the hypothesis as a start, the learner acquires additional knowledge that either supports or rejects the position. Support tends toward problem solution while rejection forces reevaluation and the generation of a new hypothesis. In contrast, data driven reasoning tends to become "analysis paralysis" rather than problem solving. In this scenario, while knowledge may be acquired, its relevance and applicability is frequently not realized.

Problem-based learning achieves its successes, in part because the experience (a) requires active processing of information, (b) activates prior knowledge, (c) provides a meaningful context, and (d) stimulates opportunities for elaboration and organization of knowledge (Barrows, 1994).

There are drawbacks to the use of problem-based learning. Teaching with problem-based learning requires a significant investment in designing problems and implementing the tutoring process (Barrows, 1988; Stepien & Pyke, 1997). For many instructors, traditional lecture formats may be both more comfortable and less effort (Bligh, 2000). Another concern found throughout the literature is that problem-based learning outcomes must not sacrifice students' learning of the subject matter. Research has shown that while problem-solving skills are better when using problem-based learning methods, simple knowledge recall of facts may be slightly less than compared with traditional methods (Vernon & Blake, 1993). However, comparisons are not always conclusive. Albanese and Mitchell (1993) in a meta-analysis of the literature reported "for six of the ten studies [comparing outcomes], the overall basic science test scores of students in conventional curricula were higher than those of students in problem-based learning curricula (negative ES); however, only three of these scores were statistically significant at the .05 level" (p. 57). They concluded that, "while the expectation that problem-based learning students will not do as well as conventional students on basic science tests appears to be generally true, it is not always true" (p. 57). In an independent meta-analysis, Vernon and Blake (1993) reached similar conclusions. Some studies found significant differences favoring traditional methods while others did not.

Relationships of PBL and SDL

While the many benefits of problem-based learning were discussed previously, the development of self-directed learners deserves special consideration. The problem-based learning literature claims the development of self-directed learners as a benefit (Barrows, 1994; Dolmans, Schmidt, & Gijselaers, 1995; Ryan, 1993; Taylor, 1986). However, these claims are not supported by research but are either theorized or postulated by supporters of problem-based learning. While the claims are not supported, neither are they refuted. The topic has not been sufficiently investigated.

A careful analysis of the literature on problem-based learning and on self-directed learning suggests definite relationships. Blumberg and Michael (1992) state that problem-based learning "... has as a primary goal the students' development of self-directed learning skills" (p. 3). However, research to determine the attainment of that goal is sparse and has taken a very narrow viewpoint. One focus has been the generation of learning issues as a measure of self-directed learning (Dolmans et al., 1995). Other research investigated the development of a few selected self-directed learning skills when a significantly teacher-directed problem-based learning method was employed (Blumberg & Michael, 1992). In the latter study, Blumberg and Michael showed that in a partially teacher-directed problem-based learning situation, problem-based learning students used the library and its resources more than traditional students, self-reported more learning resource usage, and perceived a higher proficiency in self-directed skills (Blumberg & Michael, 1992).

The more extreme, curricula-based, medical school format of problem-based learning expects a high degree of learner self-direction. For other implementations of problem-based learning, some self-directed skills are incorporated in the problem-based learning activities and usually provide more structure and scaffolding (Stepien, Senn, & Stepien, 2000; Wegner, Holloway, & Crader, 1997). Clearly, some activities used in problem-based learning require a set of skills also used in self-directed learning. This relationship suggests that using problem-based teaching methods would give students the opportunity to better develop these particular skills. Relationships between the psychological/personal characteristics of self-directed learning and problem-based learning are not obvious. The goal oriented motivational component of self-directed learning appears to coincide with the goal directedness established in problem-based learning. It is reasonable to infer that using problem-based learning might affect students' goal-oriented motivation. The problem-based learning research literature minimally addresses the development of the skills component of self-directed learning (Blumberg & Michael. 1992; Ryan. 1993).

Summary

The literature reviewed suggests that teaching with a problem-based learning method may influence students' self-directed learning. Four components of self-directed learning were identified in the literature reviewed and an integrated concept of self-directed learning was described. This literature also suggested that grades of students experiencing problem-based learning is likely to be minimally different from those taught with traditional lecture-based methods. Finally. relationships between problem-based learning and self-directed learning also suggest that teaching with problem-based learning methods is likely to affect students' self-directed learning.

CHAPTER 3

METHODOLOGY

This research compared problem-based learning instruction with traditional methods on students' self-directed learning, motivation, and grades. The three research questions were:

1. Are there significant differences between students experiencing a problem-based learning teaching method and students experiencing traditional lecture-based teaching method on:
    a. Students' self-directed learning readiness?
    b. Students' self-directed learning skills?
    c. Students' self-directed learning performance?
    d. Students' course motivation?
    e. Students' programming assignment grades?
2. Are there significant differences across time on:
    a. Students' self-directed learning readiness?
    b. Students' self-directed learning skills?
    c. Students' self-directed learning performance?
    d. Students' course motivation?
    e. Students' programming assignment grades?

3. Is there a significant interaction between teaching method and time on:

    a. Students' self-directed learning readiness?

    b. Students' self-directed learning skills?

    c. Students' self-directed learning performance?

    d. Students' course motivation?

    e. Students' programming assignment grades?

This chapter describes the research methodology. The first section discusses the design and rationale for selecting a quasi-experimental approach. Then subjects, sampling procedure, setting and materials are described. Sections on the independent variable and dependent variables provide operational definitions. The sixth major heading, "Data Collection Procedures," also includes information on treatment and measurement. Data analysis procedures follow.

Design

A quasi-experimental design was used to compare problem-based learning instruction with traditional methods on self-directed learning characteristics, motivation, and grades. The treatment group had specific problem-based learning modules taught with problem-based learning methods (the treatment) while the traditional lecture-based teaching method was used for the control group. Since students exercise their freedom of choice for a particular section of a course, random assignment of subjects was not possible.

The design for self-directed learning readiness, self-directed learning skills, motivation, and grades is illustrated in Figure 3.1. The design for self-directed learning performance lacked a pre-treatment measurement since no data was available prior to the experiment for this variable. That design is depicted in Figure 3.2. Circles represent an observation or measurement time in both figures. The X's represent treatment periods for the treatment group and the corresponding non-treatment periods for the control group. Time progresses from left to right in the figure with the groups labeled on the left.

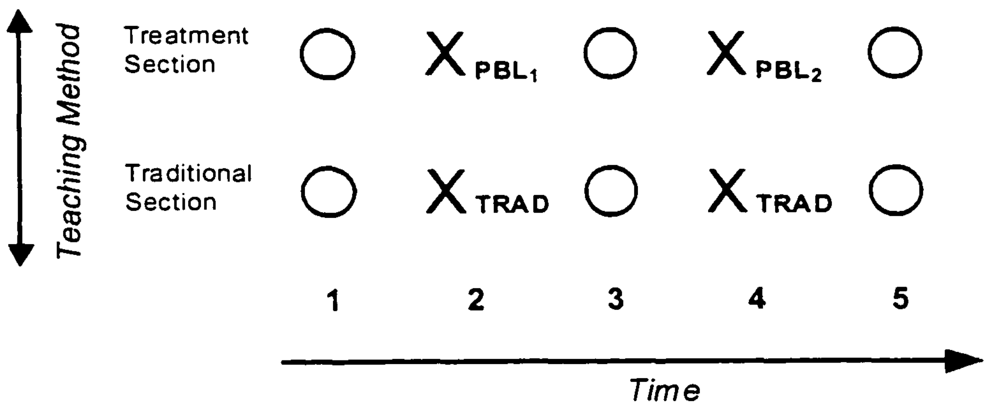

Figure 3.1. Quasi Experimental Design for SDL Readiness, Skills, Motivation, and Grades

Each treatment consisted of a single problem-based learning exercise that spanned two weeks of calendar time. Two sequential treatments were used at Times 2 and 4 illustrated in the figure. Each section of the course met twice weekly for one hour and fifty minutes. The control group met on Monday and Wednesday evenings at 7 p.m. while the treatment group met on Tuesday and Thursday evenings at 5 p.m. Pre-treatment measurements (Time 1) for both groups were made prior to the first treatment. Mid-treatment measurements (Time 3) were made after the first treatment and again at Time 5 after the final treatment.

The control group was taught using the traditional lecture-based method while the treatment group was being taught with problem-based learning. This traditional method consisted of lectures on the same topics that were identified as learning objectives used in creating the problem scenarios for the problem-based teaching. Thus. during the treatment period. the learning objectives provided by the course instructor were identical. Only the teaching method varied.

The same instructor taught both the control and experimental sections. The instructor deliberately and carefully synchronized the topics for both sections prior to and after the treatment period. Except for the problem-based learning modules. the sections were taught using the same materials. assignments. and exams. The problem-based learning instructional method used the same two programming assignments adapted from those for the control section. (The problem-based learning assignments are reproduced in A. INSTRUCTOR'S GUIDE: PBL EXERCISE 1 and C. INSTRUCTOR'S GUIDE: PBL EXERCISE 2. The corresponding control section assignments are in B. NON-PBL EXERCISE 1 and D. NON-PBL EXERCISE 2).

Self-directed learning performance data were not available prior to the beginning of the experiment since time tracking was not required prior to students' work on programming assignment #6 (the first problem-based module for the treatment group). Prior programming assignments differed in that their minimal difficulty would not have provided meaningful time tracking data on self-directed learning performance. The design shown in Figure 3.2 provided measures for self-directed learning performance representing two periods associated with the two programming assignments. No data were collected for Time 1 shown in the design. Collected data are associated with work performed during experiment Times 2 and 3 and collected upon completion of each assignment. Since students completed the assignments at varying times, the collected data only approximates collection at Times 3 and 5.

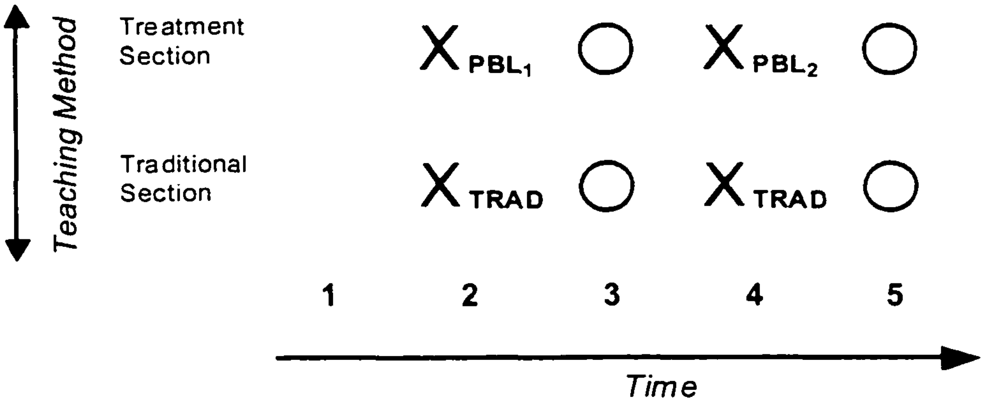

Figure 3.2. Quasi Experimental Design for SDL Performance

Subjects

The experiment was conducted with two sections of undergraduate Computer Science students enrolled in Computer Science 1 (CSI 1300) at Metropolitan State College of Denver (MSCD). These students represent typical Computer Science undergraduates needing to develop greater self-directed learning abilities. MSCD is a four-year undergraduate institution offering Bachelor's degrees. The course is described as:

> ... the first course in the computer science core sequence. Students will learn a modern programming language and the basic skills needed to analyze problems and construct programs for their solutions. The emphasis of the course is on the techniques of algorithm development, correctness and programming style. Students are also introduced to the fundamentals of software engineering and the software development life cycle (Metropolitan State College of Denver, 2001).

Students enrolled in this course were predominately undergraduate Computer Science or Mathematics majors. The sample sizes were limited to those students from each group, control and treatment, agreeing to participate in the study. Initial sample sizes consisted of 18 students (of 23 total students enrolled in the section) in the control section and 19 students (of 20 total students enrolled in the section) in the treatment group. Eight students in the control group and eight in the treatment group completed the study. These same 16 students were the only ones completing the course with a passing grade. The final sample consisted of students who generally attended class during the study, completed the study questionnaires, and completed the associated programming assignments. Students were included in the final analysis only if all three components were satisfied. Although partial data for this study were obtained on 12 students from each section (questionnaire scores for course motivation, self-directed learning skills, and self-directed learning readiness), only eight in each section completed the final programming assignment used for the post-treatment grades in the study. Only data for these 16 students was used in the final statistical analysis.

The treatment group consisted of five males and three females as compared with seven males and one female in the control group. The treatment group was more diverse than the control group having one Asian American, four Caucasian and three Hispanic students while the control group had one Asian American and seven Caucasian students.

Additional demographic data were collected to describe the sample population. The sample's mean age was 29.67. The concurrent number of courses students were enrolled in was 2.93 while the number of previous Computer Science courses was 1.87. The average total college credits of the sample were 64.27. The weekly work hours ranged from zero to 55 with a mean of 29.9 for the entire group. Student's grade point averages were 3.25 for the treatment group and 3.08 for the control group. Data for each group are shown in Table 3.1.

Table 3.1. Demographic Data

	Treatment Group			Control Group		
	Mean	Std. Dev.	n	Mean	Std. Dev.	n
Age	26.50	7.82	8	31.88	10.15	8
Num Classes	3.25	.89	8	2.63	1.06	8
CSI courses	1.25	1.16	8	2.38	1.41	8
Total Credits	62.43	40.11	8	65.88	44.30	8
Work HRS	36.07	15.92	8	24.50	17.94	8
GPA	3.25	.67	6	3.08	.89	4

Setting and Materials

The setting was a standard "smart" classroom on the college campus. Smart classrooms are equipped with LCD projectors for displaying computer output, document cameras for projecting text materials, and overhead projectors for displaying transparencies. This particular classroom configuration was used for both sections. The room has four front-facing rows of tables that each seat approximately eight students. Each seat has power and Internet connections for students with laptop computers. However, students seldom used laptops during class. The instructor routinely used all three types of media, computer, projected textbook pages, and overhead transparencies, during classes. Example source code prepared by the instructor was also frequently provided to students in both sections. These materials were learning resources complimenting the textbook.

The problem-based method used two programming assignments as the "problem" focus for teaching. Specific content learning objectives defined by the instructor for a segment of the course were incorporated in the design of the problems for the problem-based learning treatment. See A. INSTRUCTOR'S GUIDE: PBL EXERCISE 1 and C. INSTRUCTOR'S GUIDE: PBL EXERCISE 2. These materials also included information for using the problem-based learning teaching method along with guides for facilitation/tutoring the process. The problem-based treatment began with the sixth week of the semester. The corresponding two programming assignments for the control section are in B. NON-PBL EXERCISE 1 and D. NON-PBL EXERCISE 2. The teaching technique for the control group was the standard lecture-based format previously used for both sections of the course.

Independent Variable

In this study, the teaching method was the independent variable with two levels. One level was the problem-based learning teaching technique while the second level was the traditional lecture-based teaching method.

Problem-Based Learning Teaching Method

The central element of problem-based learning is a problem that is carefully selected to meet specific learning objectives. including content knowledge areas. Students in problem-based earning are expected to define their learning objectives. with significant guidance of the tutor/facilitator. as they explore the problem. This teaching method a) uses an authentic. ill-structured problem as the focal point for study. b) follows a specific process (discussed below) for investigation and inquiry. and c) is facilitated with an emphasis on the processes for inquiry and learning rather than merely providing out of context subject matter content. Furthermore. specific content knowledge related to the problem's learning objectives is not usually presented prior to the problem but is "discovered" by the students while seeking solutions to the problem. Students determine the knowledge needed to solve the problem. define and use resources to develop solutions. and review their performance of both acquiring knowledge and following the problem-solving process. This approach is contrasted with first learning and then applying knowledge.

For this study, the course instructor was the primary subject matter expert while the researcher was the primary problem-based learning tutor/facilitator for the treatment class. The researcher and instructor collaborated to maximize the quality of the in-class tutoring. Carefully scripted problem-based learning guides were used to provide scaffolding and instruction on the problem-based learning (see Problem Logs included with each problem-based learning module, A. INSTRUCTOR'S GUIDE: PBL EXERCISE 1 and C. INSTRUCTOR'S GUIDE: PBL EXERCISE 2).

Traditional Lecture-Based Learning Teaching Method

In the traditional lecture-based method, the instructor introduced topics pertinent to the learning objectives defined for a particular module. The lecture format was used, often accompanied with PowerPoint slides provided by the textbook authors. Lectures were also based on source code examples used to illustrate learning objectives. Any student questions related to the lecture were answered when posed. Explicit programming assignments with the same due dates as those for the problem-based learning group were given to the control group students. Assignments were made on the first class of the same week for each group. The control group assignments are reproduced in B. NON-PBL EXERCISE 1 and D. NON-PBL EXERCISE 2). Figure 3.3 provides a calendar of the events for the experiment.

Dependent Variables

There were three research questions addressing the teaching method, the differences over time, and the interaction of method and time. For each question five dependent variables were identified. The following discussion includes the operational definition of each dependent variable as well as its measurement method.

Self-Directed Learning Readiness

The Guglielmino Self-Directed Learning Readiness Scale (SDLRS) (Guglielmino, 1977) was used to measure self-directed learning readiness. This instrument includes 58 items using a 5-point Likert response scale. Factor analysis of the SDLRS provides eight characteristics of self-directed learners (Guglielmino, 1977). These are:

1. Openness to learning opportunities
2. Self-concept as an effective learner
3. Initiative and independence in learning
4. Informed acceptance of responsibility for one's own learning
5. Love of learning
6. Creativity
7. Positive orientation to the future, and
8. Ability to use basic study skills and problem-solving skills

Guglielmino asserted that these factors correlate favorably with the definition of a highly self-directed learner as defined by the Delphi survey of the experts. She found that "... the SDLRS could account for 76% of the variance in effectiveness as a self-directed learner" (Guglielmino, 1977, p. 73).

McCune (1988) found the SDLRS to be the most widely used instrument for measuring self-direction in learning research. SDLRS scores have shown a relatively high validity when used as a measure of readiness for self-directed learning (Bonham, 1991; Finestone, 1984; Guglielmino, 1997; Long, 1987; Long & Agyekum, 1983, 1984). Subsequent literature indicates its continued use (Confessore & Confessore, 1992a). In addition to its widespread use, research also supports its ability to indicate levels of self-directed readiness. Guglielmino (1997) cited Borg & Gall (1989) and Mehrens & Lehmann (1984) as stating the "... expert judgment is commonly used to ascertain whether an instrument has content validity" (p. 213). The Delphi technique used by Guglielmino relied on the experts on self-directed learning to provide specific topics upon which the instrument was based.

Many studies (Finestone, 1984; Hall-Johnsen, 1985; Hassan, 1981; Jones, 1989) successfully correlated scores on the SDLRS with behaviors consistent with concepts of self-directed learning. Hall-Johnsen (1985) found a positive, predictive relationship between the number of self-planned projects conducted and the time spent on these with SDLRS scores. She found that self-concept as an effective, independent learner was identified as the readiness factor that best predicted the number of self-planned projects (R = .20) and the time spent on them (R = .42). She also reported that at least five individual items on the SDLRS appear to be very effective (r = 1.00) in predicting extent of involvement in self-planned projects. Studies conducted by Finestone (1984), Hassan (1981), and Jones (1989) demonstrated validity by successfully correlating behaviors such as initiative, acceptance of responsibility for learning, and a strong desire to learn with SDLRS scores (the Pearson product-moment correlation was .48 (p =.0179)).

Reliability studies of the SDLRS have reported high Cronbach alpha estimates. Chronbach-alpha coefficient values of .87 (Guglielmino, 1977), .87 (Hall-Johnsen, 1985; Hassan, 1981), and .92 (Finestone, 1984; Skaggs, 1981) support the reliability of the SDLRS. Another reliability estimate based on a sample of 3,151 individuals from a wide variety of settings throughout the United States and Canada had the highest reported Chronbach-alpha coefficient at .94 (Guglielmino, 1989).

Self-Directed Learning Skills

The Motivational Strategies for Learning Questionnaire (MSLQ), Part B measured self-directed learning skills. Part B of the MSLQ defined learning strategies as Cognitive & Metacognitive Strategies and Resource Management Strategies. This is a self-report instrument consisting of 50 items that use a 7 point Likert scale ranging from 1 = "not at all true of me" to 7 = "very true of me." See G. MSLQ .

The category of Cognitive & Metacognitive Strategies consists of five sub categories of a) rehearsal, b) elaboration, c) organization, d) critical thinking, and e) metacognitive self-regulation. The resource management strategies consist of a) time & study environment, b) effort regulation, c) peer learning, and d) help seeking. These are elaborated in Table 3.2. MSLQ – Part B Learning Strategies.

Table 3.2. MSLQ – Part B Learning Strategies

Learning Strategy	Description
Cognitive & Metacognitive Strategies	
1. rehearsal	Reciting or naming items to be learned – influences the attention and coding process
2. elaboration	Paraphrasing, summarizing, creating analogies, and generative note-taking – helps the learner integrate and connect new information with prior knowledge
3. organization	Selection of appropriate information and construct connections – active, effortful endeavor resulting in close involvement with the task
4. critical thinking	Application of previous knowledge to new situations to solve problems, reach decisions, or make critical evaluations
5. metacognitive self-regulation	Awareness, knowledge, and control of cognition – focus is on planning, monitoring, and regulating cognitive activities
Resource Management Strategies	
1. time & study environment	Management and regulation of time and study environment
2. effort regulation	Regulation of effort related to learning goals and application of other learning strategies
3. peer learning	Realization of benefits of collaboration for increased comprehension and development of new insights
4. help seeking	Recognition of deficiencies and the implementation of strategies to define, seek, and utilize resources

Adapted from (Pintrich et al., 1991).

The skills frequently discussed throughout the literature on self-directed learning are well represented by the items addressed by the learning strategies of the MSLQ-Part B (Candy, 1991; Cheren, 1983; Grow, 1991a; Hrimech, 1995; Knowles, 1975).

Unlike the SDLRS that was designed specifically to measure a construct of self-directed learning, the MSLQ was not specifically created for self-directed learning. The correspondence of self-directed learning skills with the learning strategies items of the MSLQ – Part B provides a reasonable argument for using this instrument as a measure of the skills component for self-directed learning. The instrument itself has been shown to "... represent a coherent conceptual and empirically validated framework for assessing student motivation and use of learning strategies in the college classroom" (Pintrich, Smith, Garcia, & Mckeachie, 1993, p. 810). Their internal consistency and reliability analyses found a "... relatively good reliability in terms of internal consistency" and two confirmatory factor analyses indicated, "The general theoretical framework and the scales that measure it seem to be valid" (p. 810).

Self-Directed Learning Performance

Self-directed learning performance is defined as the extent to which students demonstrate the following performances in relation to the learning project. Knowles (1975)defined these as:

- Setting learning objectives,
- Identifying deficits in one's own knowledge in relation to the learning objectives,
- Identifying resources to address the deficits,
- Using resources for learning, and
- Self-assessment of learning outcomes.

These activities were reported through students' journaling using a modified version of "time and activity logging" developed for software programming course activities (Humphrey, 1997, pp. 21-9). Modifications that specify the types of activities provide the ability to track time and effort on self-directed learning performance related to the project. See H. PSP TIME . Students used "Engineering Notebooks," provided to both groups specifically for this study, to record their time on each task and activity. Time logs were collected weekly from each group. The same log forms were used to collect the time/performance data for both sections of the course.

Students were asked to record all time, by specific activity, spent on each of the two assignments during the experiment. The time entry forms included detailed descriptions of each activity code to assist in logging the correct activity and the time spent for that activity. The self-directed learning activities in the above bulleted list each corresponded to an activity code on the time logs. These data were collected weekly and entered into the Excel spreadsheet by activity code. The total times and the total of all self-directed learning times were then checked for correctness and imported into SPSS for analysis. The self-directed learning time represented a single dependent variable.

Motivation

The Motivational Strategies for Learning Questionnaire (MSLQ), Part A was used to measure course motivation. This part of the MSLQ has 31 self-report, Likert type items in the same format as Part B. The three primary constructs measured by this instrument are a) value, b) expectancy, and c) affective elements of motivation. The value component consists of intrinsic goal orientation, extrinsic goal orientation, and task value. Expectancy includes both control of learning beliefs and self-efficacy for learning and performance. Finally, the affective component is manifested as test anxiety. The internal consistency is high for the motivational scales with reported coefficient alphas of .90 for task value and .93 for self-efficacy for learning. Test anxiety and intrinsic goal orientation values were .80 and .74 respectively while the extrinsic goal orientation at .62 and control of learning beliefs of .68 showed more variability (Pintrich et al., 1993). Since course motivation is an integral component of student's self-directed learning (discussed in the conceptual framework section of

Chapter 1), the MSLQ – Part A was an ideally suited instrument for measuring this construct.

Grades

Grades are operationally defined as the grade assigned by the instructor on programming assignments for the related subject matter content. The grades used were the programming assignment grades prior to the treatment group's first assignment, the programming assignment grade associated with the first problem-based learning experience (programming assignment #6), and the grade for the assignment associated with the second experience (programming assignment #7). The design of the problem-based modules was such that the completed assignment's output should be identical to that of the control group's assignment's output. This allowed the instructor to grade both group's programs with the same criteria. All grades were reported on a scale from 0-100%.

Procedures

Data collection consisted of administering the SDLRS, MSLQ (Part A and B), collecting individual student activity-time logs (journals), and obtaining programming assignment grades from both groups. These data were collected prior to commencement of problem-based learning for the treatment group, between the first and second treatments, and after the second and final treatment.

The demographic data were collected concurrently with the pre-treatment administration of the other instruments. Students in the treatment group and the control group were subjected to the same measurement activities on corresponding first class meetings of the measurement week (the control group on Monday night's class and the treatment group on Tuesday night's class). The actual calendar of events is shown below in Figure 3.3. The questionnaires were administered during experiment weeks 1, 4, and 7 while time logs were collected each week. The first in-class problem-based learning treatment took place during weeks 1 and 2 with the introduction of programming assignment #6. The second in-class treatment began with programming assignment #7 in week 4 and extended through week 5. Students continued to work on the programming assignments and experience problem-based learning through week 8.

	Mon	Tue	Wed	Thur	Fri
Week 1 (Semester Week 5)	9/24 **Recruit** Explain Time Logs Handout Q1 Assign Program #6	9/25 **Recruit** Explain Time Logs Handout Q1 Assign Program #6	9/26 Collect Q1	9/27 Collect Q1	**PBL Ex 1 Program #6**
Week 2 (Semester Week 6)	10/1 Collect Time Logs #6-1	10/2 Collect Time Logs #6-1	10/3	10/4	
Week 3 (Semester Week 7)	10/8 Collect Time Logs #6-2 **EXAM #1 Prep**	10/9 Collect Time Logs #6-2 **EXAM #1 Prep No PBL**	10/10 **EXAM #1**	10/11 **EXAM #1 No PBL**	
Week 4 (Semester Week 8)	10/15 Handout Q2 Assign Program #7 Collect Time Logs #6-3	10/16 Handout Q2 Assign Program #7 Collect Time Logs #6-3	10/17 Collect Q2	10/18 Collect Q2	**PBL Ex 2 Program #7**
Week 5 (Semester Week 9)	10/22 Collect Time Logs #7-1 Collect Time Logs #6-4	10/23 Collect Time Logs #7-1 Collect Time Logs #6-4	10/24	10/25 **PBL Teaching Ends**	**Program #6 Due**
Week 6 (Semester Week 10)	10/29 Collect Time Logs #7-2 Collect Time Logs #6-5	10/30 Collect Time Logs #7-2 Collect Time Logs #6-5	10/31	11/1	
Week 7 (Semester Week 11)	11/5 Handout Q3 Collect Time Logs #7-3	11/6 Handout Q3 Collect Time Logs #7-3	11/7 Collect Q3	11/8 Collect Q3	**Program #7 Due**
Week 8 (Semester Week 12)	11/12 Collect Time Logs #7-4	11/13 Collect Time Logs #7-4	11/14 **EXAM #2**	11/15 **EXAM #2 Experiment Ends**	

□ Control Section

▒ Problem-Based Learning Section

Figure 3.3. Calendar of Events

Data were collected in paper format, transcribed into an Excel Workbook, scored using spreadsheet computations, and checked for computational and transcription errors. Upon completion of data collection, the Excel file was imported for analysis into the software program, Statistical Package for the Social Sciences (SPSS 10.05 for Windows, 1999).

Data Analysis Procedures

This study involves a single independent variable, the instructional method of a problem-based learning method versus the traditional lecture-based teaching method and multiple dependent variables. The dependent variables are assumed to be interval level. An appropriate statistical technique is an analysis of variance (ANOVA)(Cohen & Reese, 1994; Hair, Anderson, Tatham, & Black, 1998; Hertzog & Rovine, 1985; Krzanowski, 2000). When several measurements are taken over time from the same respondent, a repeated measures analysis is needed. Therefore, the repeated measurement of the same students' self-directed learning readiness, skills, performance, course motivation, and grades mandates a repeated measures ANOVA (Hair et al., 1998; Hertzog, 1994; Krzanowski, 2000; O'Brien & Kaiser, 1985; Tabachnick & Fidell, 2001).

Summary

The primary objective of this study was the determination of differences in undergraduate Computer Science students' self-directed learning after experiencing problem-based learning versus traditional instructional methods. A quasi-experimental design compared a treatment group and a control group with measurements over time of the students' self-directed traits and grades. For each of the five dependent variables, the research questions were:

1. Are there significant differences between students experiencing a problem-based learning teaching method and students experiencing traditional lecture-based teaching method?
2. Are there significant differences across time?
3. Is there a significant interaction between teaching method and time?

The five dependent variables were: 1) students' self-directed learning readiness, 2) students' self-directed learning skills, 3) students' self-directed learning performance, 4) students' course motivation, and 5) students grades on programming assignments.

CHAPTER 4

RESULTS

This study examined a problem-based learning teaching method as compared to a traditional lecture based method to determine the effects on undergraduate Computer Science students' self-directed learning, course motivation, and programming assignment grades. The independent variable was a problem-based learning teaching method versus the traditional lecture-based method. Course motivation (MSLQ-A), self-directed learning skills (MSLQ-B), self-directed learning readiness (SLDRS), self-directed learning performance (SDL task time), and grades were the dependent variables. An alpha level of .05 was used for all statistical tests.

The organization of this chapter begins with a description of the sample and the associated descriptive statistics. The next section discusses the primary data analysis including the assumptions for a repeated measures ANOVA. The "Summary of Results" section then answers each research question.

Data Analysis

The primary data analysis focused on the self-directed learning traits after students experienced problem-based learning. The statistical significance of any changes in the dependent variables over time both within the treatment group and between the treatment group and control group was determined with a repeated measures ANOVA. However, prior to this, data analysis was needed to ensure the necessary criteria were met for a repeated measures ANOVA. Each of these assumptions is addressed in the next section.

Assumptions for a Repeated Measures ANOVA

Independence. Analysis of variance assumes independent observations of the dependent variable. The repeated observation of the dependent measures in this design (pre-, mid-, and post-treatment of the experimental group) violates this assumption of independence. The repeated measures analysis compensates for the violation of this most important assumption of independence (Hair et al., 1998, p. 347).

Independence between groups was not guaranteed. However, there is evidence to suggest the observations between groups were independent. The students in each group generally lacked the opportunity to confer with those of the other group. The groups met on alternate nights, students had heavy work schedules, and were generally taking two additional classes. One student questioned: "What is the other section doing?" The response from a only one student was "the same thing." Later private discussion with the respondent revealed some awareness but no conferring on substantive topics. The policy of the instructor and researcher was to avoid. as much as possible. discussing one section with the other. Overheard comments of students also left the impression that most were not aware another section was involved in the study until after mid-way through the experiment. Thus, it was assumed their was sufficient independence between groups.

Equality of Variance-Covariance Matrices. This assumption calls for the equality of the variance-covariance matrix (Girden, 1992; Hertzog & Rovine, 1985; O'Brien & Kaiser. 1985). Violation of this assumption increases the Type I error in the main effects and interactions as well as results in a loss of power (O'Brien & Kaiser, 1985, p. 317). The Levene test of the homogeneity of variance for each dependent variable across all level combinations of the between-subjects factors determined that the error variance of the dependent variables was equal across groups with the only possible exceptions being the pre-treatment self-directed learning skills and the pre-treatment grades. See I. : Levene's Test of Equality of Error Variances. However, this problem is inconsequential since "... a violation of this assumption has minimal impact if the groups are of approximately equal size" (Hair et al., 1998, p. 348). The groups in this study were equal in size.

Sphericity. In the repeated measures analysis, all variances of the repeated measurements should be equal and all correlations between the pairs of repeated measurements should also be equal. Violations of sphericity inflates the Type I error rate. Mauchly's test for sphericity indicated that the assumption of sphericity was met (see I. : Mauchly's Test of Sphericity).

Normality. Another assumption for a repeated measures ANOVA is that the dependent measures are normally distributed. Tests of normality confirmed that all dependent variable measurements appear to be normally distributed except for the grades of the control group on programming assignment #6 and assignment #7. The Shapiro-Wilk test (see I. : Tests of Normality) was an appropriate tool because of the small sample size (SPSS, 1999)). Normal Q-Q Plots of expected normal values versus observed values exhibit linear correlations as expected for normally distributed data and corroborate the Shapiro-Wilk test. Because MANOVA is relatively robust to violations of normality (Hair et al., 1998; O'Brien & Kaiser, 1985), the non-normal programming assignment grades is less problematic.

Thus the criteria necessary for a repeated measures ANOVA were either met or determined to have little adverse impact.

Demographic Data

Demographic data for the treatment and control groups are summarized in Table 4.1. There were no statistically significant differences between the control group and the experimental group on any of the demographic characteristics.

Table 4.2 shows the results of the independent samples *t*-Tests for the metric demographic data.

Table 4.1. Demographic Data

	Treatment Group			Control Group		
	Mean	Std. Dev.	n	Mean	Std. Dev.	n
Age	26.50	7.82	8	31.88	10.15	8
Num Classes	3.25	.89	8	2.63	1.06	8
CSI courses	1.25	1.16	8	2.38	1.41	8
Total Credits	62.43	40.11	7	65.88	44.30	8
Work HRS	36.07	15.92	8	24.50	17.94	8
GPA	3.25	.67	6	3.08	.89	4

Table 4.2. *t*-Test of Demographic Data

	t-test for Equality of Means						
	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
						Lower	Upper
Age	1.19	14	.255	5.38	4.53	-4.34	15.09
Num Classes	-1.28	14	.222	-.63	.49	-1.67	.42
CSI courses	1.74	14	.104	1.13	.65	-.26	2.51
Total Credits	.157	13	.878	3.45	21.95	-43.98	50.87
Work HRS	-1.36	14	.194	-11.56	8.48	-29.751	6.626
GPA	-.36	8	.729	-.176	.49	-1.306	.955

Results of Major Analysis

The effects of the teaching method, time, and the interaction of method and time were determined for each dependent variable. The following descriptive statistics tables for each dependent variable present the means, standard deviation, and sample sizes by teaching method for each of the three measurement times. These tables also include summary statistics by teaching method and times. The ANOVA tables for each dependent variable are presented immediately after the associated descriptive statistics table. Statistically significant results are briefly noted and include plots of estimated marginal means over time for each teaching method.

Table 4.3. Descriptive Statistics for Self-Directed Learning Readiness

		TIME			
		1	2	3	
Teaching Method	Traditional	224.12 (31.98) n = 8	222.38 (30.33) n = 8	223.50 (25.63) n = 8	224.12 (31.98) n = 24
	PBL	235.63 (31.28) n = 8	232.38 (29.15) n = 8	233.38 (31.63) n = 8	235.63 (31.28) n = 24
		229.88 (31.13) n = 16	227.38 (29.20) n = 16	228.44 (28.27) n = 16	

Table 4.4. ANOVA Table for Self-Directed Learning Readiness

SV	SS	df	MS	F	p-value
Method (M)	1312.521	1	1312.521	.505	.489
s: M	36351.958	14	2596.568		
Time (T)	50.375	2	25.187	.428	.656
M x T	6.542	2	3.271	.056	.946
s: M x T	1646.417	28	58.80		

Table 4.5. Descriptive Statistics for Self-Directed Learning Skills

		TIME			
		1	2	3	
Teaching Method	Traditional	208.75 (35.65) n = 8	210.5 (47.22) n = 8	223.12 (27.81) n = 8	214.13 (33.91) n = 24
	PBL	222.00 (13.21) n = 8	214.75 (30.74) n = 8	216.88 (31.81) n = 8	217.88 (22.80) n = 24
		215.38 (26.86) n = 16	212.63 (38.55) n = 16	220.00 (29.05) n = 16	

Table 4.6. ANOVA Table for Self-Directed Learning Skills

SV	SS	df	MS	F	p-value
Method (M)	168.750	1	168.750	.067	.799
s: M	35064.583	14	2504.613		
Time (T)	444.500	2	222.250	.637	.537
M x T	762.000	2	381.000	1.091	.350
s: M x T	9774.167	28	349.077		

Table 4.7. Descriptive Statistics for Self-Directed Learning Performance

		TIME			
		1	2	3	
Teaching Method	Traditional	No Data	102.17 (52.21) n = 6	72.50 (70.99) n = 6	87.33 (56.43) n = 12
	PBL	No Data	200.00 (46.44) n = 6	142.50 (83.89) n = 6	171.25 (42.82) n = 12
		No Data	151.08 (69.50) n = 12	107.50 (82.62) n = 12	

Table 4.8. ANOVA Table for Self-Directed Learning Performance

SV	SS	df	MS	F	p-value
Method (M)	42252.042	1	42252.042	8.420	.016
s: M	50179.417	10	5017.942		
Time (T)	11397.042	1	11397.042	3.292	.100
M x T	1162.042	1	1162.042	.336	.575
s: M x T	34616.417	10	3461.642		

The effect of method on performance is significant. The effect of teaching method was significant with the problem-based learning group mean time of 171.25 as compared to the traditional group's 87.33 minutes (see Table 4.7). However, the effect of time and the method x time interaction were not significant. The possible explanations are discussed in chapter 5.

Table 4.9. Descriptive Statistics for Motivation

			TIME		
		1	2	3	
Teaching Method	Traditional	167.75 (23.73) n = 8	167.75 (24.38) n = 8	169.62 (21.87) n = 8	168.38 (21.87) n = 24
	PBL	165.38 (17.25) n = 8	159.50 (19.40) n = 8	156.88 (15.88) n = 8	160.58 (15.20) n = 24
		166.56 (20.08) n = 16	163.63 (21.70) n = 16	163.25 (18.07) n = 16	

Table 4.10. ANOVA Table for Motivation

SV	SS	df	MS	F	p-value
Method (M)	728.521	1	728.521	.685	.422
s: M	14893.458	14	1063.818		
Time (T)	105.292	2	52.646	.679	.515
M x T	216.542	2	108.271	1.396	.264
s: M x T	2172.167	28	77.577		

Table 4.11. Descriptive Statistics for Grades

		TIME			
		1	2	3	
Teaching Method	Traditional	98.000 (1.60) n = 8	84.69 (12.04) n = 8	77.97 (11.08) n = 8	86.89 (7.05) n = 24
	PBL	94.25 (4.89) n = 8	61.25 (15.32) n = 8	63.13 (18.24) n = 8	72.88 (11.92) n = 24
		96.13 (4.01) n = 16	72.97 (17.99) n = 16	70.55 (16.47) n = 16	

Table 4.12. ANOVA Table for Grades

SV	SS	df	MS	F	p-value
Method (M)	2355.501	1	2355.501	8.187	.013
s: M	4027.914	14	287.708		
Time (T)	6380.362	2	3190.181	44.585	< .001
M x T	779.362	2	389.681	5.446	.010
s: M x T	2003.484	28	71.553		

The effect of teaching method, time, and the interaction of method x time on grades was significant. The means for grades by teaching method were 86.89 for the traditional and 72.88 for the problem-based group. The means over time were 96.13, 72.97, and 70.55 for times 1, 2, and 3 respectively (see Table 4.11). The significant interaction is illustrated in Figure 4.1. Grades of both groups dropped dramatically between the first assignment (Time 1) and subsequent assignments (Times 2 and 3). The drop is much greater for the problem-based learning teaching method group. Possible explanations are discussed in chapter 5.

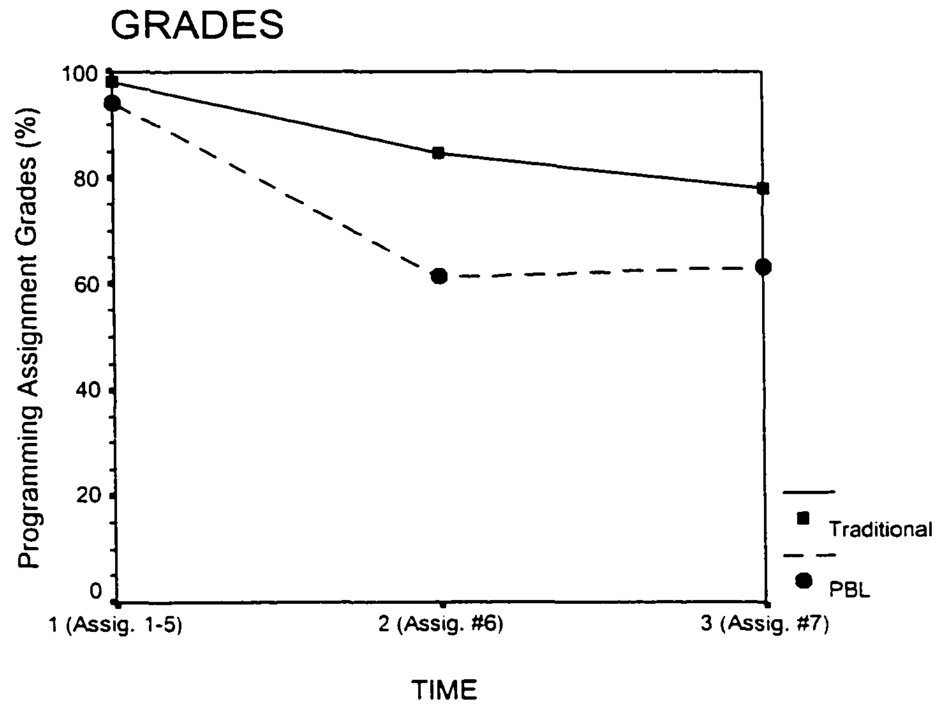

Figure 4.1. Programming Assignment Grades

Summary of Results by Research Question

This section presents the results of the statistical analyses organized by individual research question. The repeated measures analysis of variance revealed no statistically significant differences for any of the dependent variables with the exception of grades.

Research Question 1

The first question asked whether there are significant differences between students experiencing a problem-based learning teaching method and students experiencing traditional lecture-based teaching method for a) self-directed learning readiness, b) self-directed learning skills, c) self-directed learning performance, d) students' course motivation, and e) programming assignment grades.

Self-Directed Learning Readiness. There were no statistically significant differences in self-directed learning readiness scores regardless of the teaching method. The F ratio for the main effect was *F(1,14)* = .505, *p* =.489. (Table 4.4, page 64).

Self-Directed Learning Skills. There were no statistically significant differences in self-directed learning skills scores regardless of the teaching method. The F ratio for the main effect was *F(1,14)* = .067, *p* =.799. (Table 4.6, page 65).

Self-Directed Learning Performance. The effect of teaching method was statistically significant for self-directed learning performance. The F ratio for the main effect was *F(1,10)* = 8.42, *p* =.016. (Table 4.8, page 66).

Students' Course Motivation. Differences in students' course motivation were not statistically significant regardless of the teaching method. The F ratio for the main effect was *F(1,14)* = .685, *p* =.422. (Table 4.10, page 67).

Programming Assignment Grades. Differences in grades were statistically significant. The F ratio for the main effect was *F(1,14)* = 8.187, *p* =.013. The mean of the lecture-based group's grades was higher. (Table 4.12, page 68).

Research Question 2

The second question asked whether there are significant differences over the three time periods for a) self-directed learning readiness, b) self-directed learning skills, c) self-directed learning performance, d) students' course motivation, and e) programming assignment grades.

Self-Directed Learning Readiness. There were no statistically significant differences in self-directed learning readiness scores over the three time periods (pre-, mid-, or post-treatment). The F ratio for the time effect was *F(2,28)* = .428, *p* =.656. (Table 4.4, page 64).

Self-Directed Learning Skills. There were no statistically significant differences in self-directed learning skills scores over the three time periods (pre-, mid-, or post-treatment). The F ratio for the time effect was *F(2)* = .637, *p* =.537. (Table 4.6, page 65).

Self-Directed Learning Performance. There were no statistically significant differences in self-directed learning performance scores over the two times (corresponding to assignments #6, T2 and assignments #7, T3). No data were available for self-directed learning performance prior to the beginning of the first treatment period. The F ratio for the time effect was $F(1,10) = 3.292$, $p = .100$. (Table 4.8, page 66).

Students' Course Motivation. Differences in students' course motivation were not statistically significant over the three time periods (pre-, mid-, or post-treatment). The F ratio for the time effect was $F(2,28) = .679$, $p = .515$. (Table 4.10, page 67).

Programming Assignment Grades. Grades were statistically different over time within-groups. The F ratio for the time effect was $F(2,28) = 44.585$, $p < .001$. (Table 4.12, page 68). The grades of both groups declined significant from the pre-treatment time to mid- and post-treatment times.

Research Question 3

The third question asked whether the interaction of time and teaching method would be have a significant effect on a) self-directed learning readiness, b) self-directed learning skills, c) self-directed learning performance, d) students' course motivation, and e) programming assignment grades.

Self-Directed Learning Readiness. The self-directed learning readiness scores showed no statistically significant differences for the method x time interaction. The F ratio for the method x time interaction was *F(2,28)* = .056, *p* =.946. (Table 4.4, page 64).

Self-Directed Learning Skills. There were no statistically significant differences in self-directed learning skills scores for the method x time interaction. The F ratio for the method x time interaction was *F(2,28)* = 1.091, *p* =.350. (Table 4.6, page 65).

Self-Directed Learning Performance. There were no statistically significant differences in self-directed learning performance scores for the method x time interaction. The F ratio for the method x time effect was *F(1,10)* = .336, *p* =.575. (Table 4.8, page 66).

Students' Course Motivation. Differences in students' course motivation were not statistically significant for the time*method interaction. The F ratio for the method x time effect was *F(2,28)* = 1.396, *p* =.264. (Table 4.10, page 67).

Programming Assignment Grades. There was a statistically significant method x time interaction. The F ratio for the time effect was *F(2,28)* = 5.446, *p* =.010. (Table 4.12, page 68).

Summary

In summary, the effects of teaching method, time, and method x time interaction was not statistically significant on students' self-directed learning readiness, self-directed learning skills, or course motivation occurred. These self-directed learning components did not differ for the students experiencing a problem-based learning teaching method nor was there any differences for the traditional lecture-based method.

The effect of teaching method on self-directed learning performance was statistically significant with the problem-based method groups' performance scores greater than those of the traditional lecture-based groups.' However, the effect of time and the method x time interaction were not statistically significant for self-directed learning performance.

Programming assignment grades appeared to differ significantly with teaching method, over time, and with the method x time interaction. The traditional lecture-based method group consistently demonstrated statistically higher grades than the problem-based learning method group.

CHAPTER 5

DISCUSSION

A problem-based learning teaching method was compared with a traditional lecture-based teaching method to determine the effects on undergraduate Computer Science students' self-directed learning and programming assignment grades. An integrated construct of self-directed learning included a) self-directed learning readiness b) self-directed learning skills, c) self-directed learning performance, and d) students' course motivation.

Quasi-experimental designs were used to compare a problem-based teaching method and a traditional lecture-based method in two sections of a CS1 course taught by the same instructor. Each of the self-directed learning components and grades were measured for students experiencing traditional instructional methods and problem-based learning methods. Readiness was measured with the Self-Directed Learning Readiness Scale, skills with the Motivated Strategies for Learning Questionnaire–Part B, performance with time spent on self-directed learning tasks, and course motivation with the Motivated Strategies for Learning Questionnaire–Part A. The grade measurement was the course instructor's percentage score given to students' programming assignments.

The results, described in Chapter 4, revealed significant differences in performance but showed no significant differences for either group in students' self-directed learning readiness, self-directed learning skills, or course motivation. The effect of teaching method was statistically significant on problem-based learning performance with the treatment group spending more time on self-directed learning tasks. The effects of teaching method, time, and the interaction of method x time were statistically significant on grades. All grades of the group taught with problem-based learning methods were lower than those of the group taught with traditional lecture-based methods. However, the programming assignment grades of both groups significantly declined over time.

The remainder of this chapter discusses these findings, provides possible explanations for the lack of significant differencess in self-directed learning traits, examines the problem-based learning treatment, addresses limitations of the study, and offers topics for further investigation.

Self-Directed Learning Components

The conceptual framework proposed an integrated self-directed learning construct composed of self-directed learning readiness, self-directed learning skills, self-directed learning performance, and students' course motivation. For each of these dependent variables, the three research questions asked: a) Are there significant differences between students experiencing a problem-based learning teaching method and students experiencing traditional lecture-based teaching method? b) Are there significant differences across time? And c) Is there a significant interaction between teaching method and time?

Students in the both groups exhibited no significant differences in the readiness, skills, or motivation. One possible explanation that applies for each dependent variable is the low statistical power resulting from extremely small sample sizes (eight in each group). Only large effects are likely to be observed with these sample sizes.

Self-Directed Learning Readiness

The lack of an observed difference in students' scores for self-directed learning readiness could be because a) the problem-based learning had no effect, b) the effect size was too small to observe given the small sample size, c) the level, quality, or duration of the problem-based learning treatment was insufficient to have an observable effect or d) the already relatively high self-directed learning readiness left little room for increase.

The overall self-directed learning readiness scores (mean of approximately 229) were considered "above average" and just below "high." SDLRS scores are categorized as Low (58-188), Below Average (189-203), Average (204-218), Above Average (219-232), and High (233-290) with an overall population mean of 214 (Jones, 1989).

Table 4.3 (page 64) shows that the problem-based learning groups' readiness scores were consistently higher over time although the difference is not statistically significant (see Table 4.4, page 64). The problem-based group would be categorized as "high" in self-directed learning while the traditional lecture-based group remained in the "above average" range. Both groups are certainly above average, although not statistically different from each other.

There are several possible explanations for no significant differences in SDLRS scores. In a study using learning contracts as a tool to teach self-direction, Caffarella and Caffarella (1986) found no differences in SDLRS scores of students measured at the start and end of the course. They did find some, although limited, impact on self-directed learning. Two of their conclusions may be pertinent to the findings in this study. The SDLRS measures attitudes towards self-directedness rather than specific abilities so differences in competencies may not be reflected in differences in attitudes. The ceiling effect may also be a factor in the lack of differences. Caffarella and Caffarella (1986) argued that high initial scores on the SDLRS leave little room for significant increases. Although the scores of the CS1 undergraduates were in the mid 220's to mid 230's as compared to the graduate students' scores at 240 in the Caffarella study, the ceiling effect may have played some role here too.

Examination of the scores in Table 4.3 (page 64) reveals that students were very stable in their attitudes toward self-directed learning readiness. The students' report of their perceptions of self-directed learning readiness did not change.

Self-Directed Learning Skills

The failure to find a difference in self-directed learning skills can be attributed to most of the same factors as those for self-directed learning readiness: a) the problem-based learning had no effect, b) the effect size was too small to observe given the small sample size or c) the level, quality, or duration of the problem-based learning treatment was insufficient to have an observable effect. Table 4.5 (page 65) shows that the problem-based learning groups' skills scores were initially higher but declined and remained essentially flat over time although the difference is not statistically significant (see Table 4.6, page 65). However, the traditional groups' scores revealed an increase for the last measurement. Again, these differences were not statistically significant.

The possibility remains that the MSLQ-B does not measure the exact skill set required for self-directed learning. In Table 3.2 (page 25), the skill sets for the MSLQ-B include many skills important to self-directed learning but these may not represent all the specific skills required. The assessed skills may be necessary but not sufficient to fully describe self-directed learning skills. Although beyond the scope of this study, correlations of subcategories from the instrument with self-directed learning might prove useful. Future research is needed to investigate specific measures of self-directed learning skills both as perceived by the student and as demonstrated.

Self-Directed Learning Performance

The analyses for self-directed learning performance showed that the effect of teaching method was significance at the .05 level. The F ratio for the teaching method effect was $F(1, 10)$ = 8.420 p =.016 (see Table 4.8, page 66). However, the effect for time was not significant (F ratio for the time effect was $F(1, 10)$ = 3.292, p =.100). The method x time interaction was also not significant with an F ratio of $F(1, 10)$ = .336, p =.575. (Table 4.8, page 66). The following observations are made with the recognition that the sample size was small. No data were available on self-directed learning performance time prior to the beginning of the experiment so only time on assignments #6 and #7 were available. The prior programming assignments were not sufficiently complex that time tracking by the activity codes would have been meaningful. Description of these pre-treatment assignments is more fully discussed under "Grades," page 82.

Table 4.7, page 66, shows the significantly higher performance times for the problem-based learning group. The use of the problem-based learning teaching method required students to spend more time on the self-directed learning activities. Without the guidance and structure of the problem-based learning method, the lecture-based group reported less time thinking about their learning needs, how to address them, and reflecting on their own learning than did the problem-based group.

The declines in self-directed learning performance over time shown in Table 4.7 maybe explained by the need for less time to complete the second assignment. The second assignment allowed some reuse of skills and knowledge necessary for the first assignment. Both groups required less total time to complete the second assignment (only 55% of the time needed for the first assignment for the treatment group and 68% for the control group).

There were concerns about the quality of the reported time data. Students were asked to keep time logs for all their activities associated with each programming assignment. All time spent on each assignment should have been designated with an activity code designed to identify self-directed learning performance. Initially, students diligently recorded their time and activity codes. However, students reported difficulty in accurately partitioning time into appropriate activity codes. Some students may have given up accurately reporting correct activity codes.

The quality of the time reporting data and inadequate sample size reduce the confidence in these findings. Further discussion of the performance component is found in recommendations for future research.

Course Motivation

Problem-based learning and traditional teaching methods were not shown to have a significant effect on the motivation component. The problem-based learning group experienced an insignificant decline in motivation scores over time (shown in Table 4.9, page 67). The use of authentic problems in the problem-based learning method did not appear to impact student motivation. The students' experience of a new, unfamiliar teaching method may account for the lack of effect. Although beyond the scope of this study, an examination of the individual motivational components of the MSLQ-A might offer greater insight.

A final consideration is that the direct connection between motivation measured by the MSLQ-A and motivation for self-directed learning may be too amorphous to yield meaningful results. However, since differences in motivation scores were not statistically significant in this study, further discussion is not warranted.

Grades

The statistical analysis showed that the effect of teaching method, time, and the interaction of method x time on grades was significant. The problem-based method group consistently earned lower grades over time for the programming assignments than the traditional method group. The obvious conclusion is that a traditional lecture-based teaching method yields better grades. However, additional investigation revealed other factors that likely influenced grades. Although analysis of the demographic data showed no statistical differences between the groups, some of these factors may have contributed to lower grades for the treatment group.

Both group's pre-treatment grades were extremely high compared to their mid- and post-treatment scores. These declines and the differences between the groups are dramatically illustrated in Figure 5.1 below.

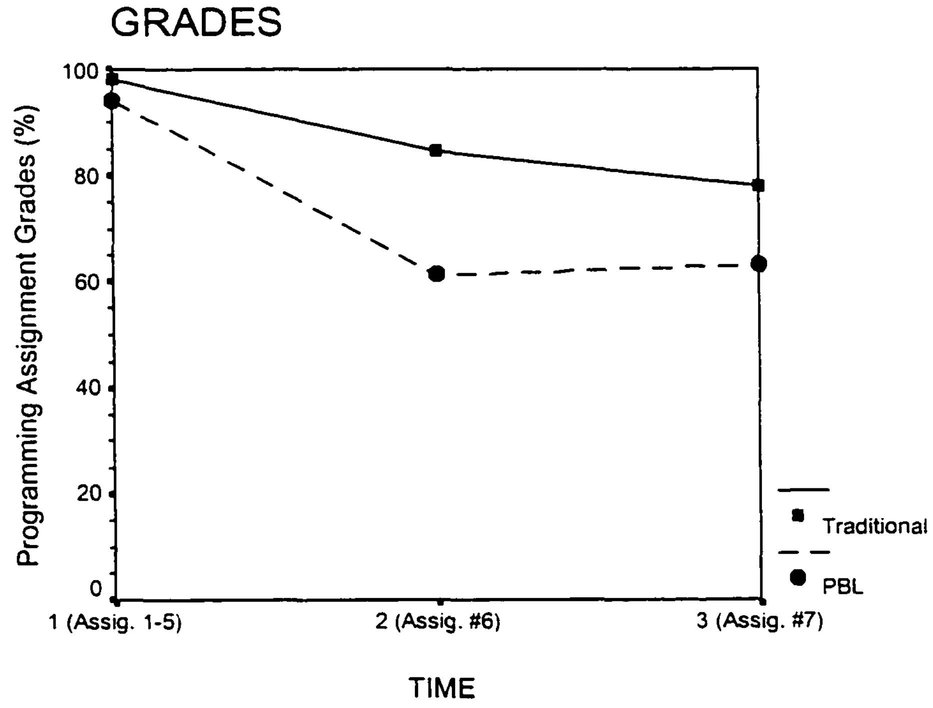

Figure 5.1. Programming Assignment Grades

The average grades on the pre-treatment programming assignments were 98% for the control group and 94.25% for the treatment group. These anomalously high pre-treatment grades can be accounted for by two factors. First, the pre-experiment assignments lacked the higher level of difficulty found in assignments #6 and #7. The second factor is that the grading scheme prior to the experiment was more lenient than that for the experiment assignments.

The first assignment score consisted of an aggregation of short assignments requiring the students to "type and run" programs. Students were given paper copies of simple programs and code components of programs from which they created their own program. These tasks were primarily a test of their ability to configure their programming environment rather than define, design, and implement a program. The assignments for programs #6 and #7 were significantly more difficult requiring students to define, design, and implement a solution to the problem on their own.

The grading scheme varied between the first assignment and subsequent assignments. Students were allowed to submit multiple times with instructor feedback each time for the first assignment before the final grade was assigned. However, the grades for assignments #6 and #7 were based on a single submission of the student's program without prior instructor feedback.

The change in assignment difficulty and the more stringent grading of the last two assignments explains the lower grades as compared to the first assignment. Since both groups experienced the drop, it is difficult to attribute the difference over time primarily to teaching method.

The treatment group's consistently lower grades may also have been influenced by factors other than, or in addition to, teaching method. Compared to the traditional method group, students in the problem-based group had completed fewer computer science courses, worked more hours outside of school each week, carried a heavier concurrent course load, and were younger. Table 5.1 compares these factors for the two groups. While none of these differences were statistically significant, the influence on programming ability could contribute to differences in grades.

Table 5.1. Selected Demographic Data

	PBL Group Mean	Traditional Group Mean
Previous CS Courses	1.29	2.38
Work HRS peer Week	36.07	24.50
Number Concurrent Classes	3.29	2.63
Age	27.14	31.88

The actual experience level of programming expertise for the control group seems to have been greater with an average of one more course than the treatment group. The control group also spent 11.5 hours less working each week and was taking less other courses than the treatment group.

Problem-Based Learning Teaching Method

Since the research was directed toward comparing problem-based learning instruction versus traditional lecture-based instruction on students' self-directed learning. the problem-based method is detailed here. The problem-based learning treatment involved two problem scenarios associated with programming assignments (See A. INSTRUCTOR'S GUIDE: PBL EXERCISE 1 and C. INSTRUCTOR'S GUIDE: PBL EXERCISE 2). Each problem was presented. investigated. and studied over two weeks class time (4- two hour class periods) using the problem-based learning teaching method. The actual calendar time included an additional week between the two problem-based learning experiences during which the instructor gave a review session class and the first exam of the semester. Instructor imposed deadlines for the programming assignments were an additional two weeks beyond the completion of the problem-based learning experience. The actual sequence of events is shown in Figure 3.3. Calendar of Events (page 55).

During the problem-based learning treatment of the experimental group. the instructor served as a subject matter expert and co-tutor while the researcher functioned as the primary problem-based learning tutor. The Problem Logs 1 – 9 of the instructor's guides for PBL were used to ensure the problem-based learning experience was administered consistently and correctly (See A. INSTRUCTOR'S GUIDE: PBL EXERCISE 1 and C. INSTRUCTOR'S GUIDE: PBL EXERCISE 2 for Problem Logs 1 – 9). The problem-based learning method included the characteristics described in Table 5.2.

Many of the process steps required students to individually complete the activities begun in class (especially sequence steps 2, 3, and 4). The continuation of the process outside of class and the engagement of students in-class were problematic. Deficits in the overall quality of the problem-based learning treatment received by the students warrant further discussion.

Table 5.2. Problem-Based Learning Characteristics

Characteristic	Description
Problem	Ill-defined & complex
Learning Issues	Teacher defined through selection of problem though not explicitly revealed to students; Student-defined with facilitation toward problem solution
Tutoring/Facilitation	Tutor (the researcher) facilitated the problem-based learning process; Instructor as a subject matter expert resource and secondary tutor; Scripted guides for the instructor/tutor and guided exercises for the students
Group size	Average size of 4-5 for problem clarification and definition of learning issues; process tutoring and plenary sessions in both groups and as a whole class
Process sequence	1. Problem presentation 2. Groups refine problem aspects and define needed learning issues (in-class) with facilitation (what they know, what they need to know) 3. Groups define resources (in-class) with facilitation 4. Individuals use resources for learning 5. Individuals share knowledge in groups (in-class) 6. Groups summarize results for learning issues (in-class) with facilitation 7. Individuals complete implementation of problem solution
Problem duration	Approximately 2 weeks (calendar time)

Quality of Problem-Based Learning Treatment

Although the teaching method was rigorously followed, student reception, responsiveness, and participation varied. Observations during the experiment suggested that the quality of the treatment experienced by individual students varied greatly.

Some students remained fairly passive in spite of the active learning teaching activities. If students failed to engage or participate either in-class or outside of class, the treatment was diminished. The recalcitrant students also failed to demonstrate basic study behaviors such as note taking, reading the textbook, reading instructor prepared handouts, or reading the problem-based learning guides.

Significant lack of regular attendance also diminished the treatment for all students. Even fully engaged students were adversely affected by others' disengagement. Marginally engaged and absent students disrupted many of the collaborative activities in problem-based learning. In addition to irregular attendance, lack of in-class participation and failure to bring new knowledge for sharing with peers (step 5 in the process sequence) were two commonly observed problems. These problems negatively affected the quality of the experience since problem-based learning requires a significant collaborative learning component (Cockrell et al., 2000; Hmelo & Ferrari, 1997). Successful collaborative learning requires positive interdependence among group members (Johnson & Johnson, 1999) which is difficult to achieve with such dynamic groups. Unfortunately, participating students' experience was adversely impacted by their peers' failure to engage, participate, or attend regularly.

Another adverse factor that may have lessened the quality of the treatment is the transition from the familiar, traditional lecture-based teaching to the problem-based method. Student difficulties in transitioning to a different, more active, teaching method, specifically problem-based learning, have been noted by others (Dunlap, 1996; Loats, 2001). The use of the problem-based learning scripts and logs was used as scaffolding to facilitate this adjustment and insure the process was accurately implemented (A. INSTRUCTOR'S GUIDE: PBL EXERCISE 1 and C. INSTRUCTOR'S GUIDE: PBL EXERCISE 2, Problem Logs 1 – 9). These scripts specifically addressed this concern of needing to learn the process. The addition of a second treatment period also allowed students multiple experiences to reach a certain level of expertise with the process.

The lack of participation during in-class group activities was observed and efforts were made to rectify the problem. A significant portion of the tutorial effort focused on engaging non-contributing students. The reluctant students were gently coerced toward a perfunctory level of engagement with in-class activities. However, time logs indicated little or no outside of class effort from these students. Many of these students reported zero task time on outside activities during the weeks of each problem-based learning module. These students' time reporting revealed the effort was greatest immediately prior to assignment due dates. Table 5.3 shows the number of students reporting time (greater than zero minutes) on the assignment for each week. Only students completing the study are included. The number of students reporting times greater than zero reveals that the treatment group students apparently did not report all their time as the experiment progressed. Two students submitted assignments after week 7 without reporting any time after week 7.

Table 5.3. Reported Total Effort (Time) on Assignments by Week

Week	Activity	Program #6				Program #7			
		Treatment Group		Control Group		Treatment Group		Control Group	
		N[1]	Time[2]	N[1]	Time[2]	N[1]	Time[2]	N[1]	Time[2]
1	PBL Treat Program #6		201 (8)		141 (5)				
2			165 (8)		354 (4)				
3	Exam 1		184 (4)		175 (2)				
4	PBL Treat Program #7		307 (5)	1	239 (4)		166 (7)		261 (3)
5		2	462 (5)	6	439 (7)		321 (2)		290 (1)
6	#6 Due	1	163 (3)	1		1	201 (6)		285 (7)
7		3	425 (1)			3	493 (6)	5	267 (5)
8	#7 Due					2		1	
9						1			
10		1							
11		1				1			

[1] Number of students submitting the assignment
[2] Average weekly times in minutes reported. Number of students reporting time > 0 is in parentheses.

The extended due dates for the programming assignments also allowed the less engaged and procrastinating students to dissociate the problem-based learning experience from the actual problem solving (and learning) necessary to complete the assignment. Due dates for the assignments extended an additional two weeks beyond the completion of the associated experience (see calendar Figure 3.3, page 55). Most students focused on completing the programming assignments well after the in-class problem-based learning experience. The treatment group reported that over 83% of the time on program #6 was expended after the associated problem-based learning was completed (see Table 5.3 above). The earliest submitted work for the treatment group was in the third week after the completion of the corresponding problem-based learning activity. Three students submitted in week 7, fully three weeks after the experience while two other submissions came more than eight weeks later.

As with any real-world classroom, the teaching method (treatment) was received and experienced to differing degrees by each student. No attempt was made to measure these levels other than to observe the phenomenon during the experiment. It is possible that the overall treatment level may have been minimal for many of the students. The students unevenly experienced the problem-based learning treatment because of irregular attendance, inadequate participation, and lack of concurrency of treatment with problem solution. The acceptance and participation in the treatment was analogous to patients electing to take all, part of, or none of the prescribed medication.

Limitations

The generalizability of these results to a larger population cannot be made because of a very small sample size. Also, the use of only one class in each group seriously limits the generalizability of the findings. The nature of teaching with problem-based learning in an undergraduate Computer Science setting generally precludes very large sample sizes. Unfortunately, in this study, sample size was further reduced by the extremely high attrition. The study began with nearly 40 subjects and ended with only 16. Too many students simply failed to complete the necessary coursework. For the two sections, the course failure rate was 55%. Furthermore, since there were only two sections of a single course used for the experiment, there was a selection threat to internal validity (Campbell & Stanley, 1963).

A second limitation was the necessary assumption that students honestly and diligently completed the questionnaires. Based on observations, students did not necessarily place a high priority on carefully and thoughtfully completing the questionnaires. Those who forgot to complete them before class would quickly complete them in the time between their arrival and the start of class.

Another limitation was the type of instruments used to measure self-directed learning readiness, skills, and motivation. Each instrument is a self-report of the perceptions of the individual. These instruments do not measure actual self-directed learning skills, performance/behaviors, or motivation. Students may perceive they possess a higher capability than they actually demonstrate. Furthermore, students can easily discern which response on the Likert type scale is considered "better" even though instructions ask for their honest perceptions, not what they think are correct or better answers.

The adaptation of the MSLQ-B to measure self-directed learning skills may also be a limitation since it may not measure all the skills necessary for self-directed learning. While it provides a measure of learning skills that are necessary for self-directed learning, there may be other critical self-directed learning skills that are not represented.

Recommendations for Future Research

This research compared a problem-based learning teaching method with a traditional lecture-based method to determine the effect on students' self-directed learning and grades. Several future research avenues for both problem-based learning and self-directed learning are apparent from this study.

It appeared that many students in the experimental group did not adequately participate in the problem-based learning activities. Further research should assess the actual level and quality of problem-based learning experienced by students. Measuring students' ability to practice problem-based learning would be a positive addition to our understanding of the experience. Most of the existing literature in this area focuses on assessing learning outcomes (e.g. subject matter knowledge and development of problem-solving skills) of students taught with problem-based learning (Dathe. O'Brien, Loacker, & Matlock, 1997; Norman, 1997; Segers, 1997; Swanson, Case, & van der Vleuten, 1997). Measures of the quality and level of students' practice of problem-based learning might be correlated with learning outcomes.

The need for students to acclimate to a different teaching method such as problem-based learning further suggests additional research. A longer duration for problem-based learning and corresponding research might provide more useful data. Students accustomed to passive learning are not immediately comfortable or competent with active, collaborative activities.

Another need for future research is to distinguish self-directed learning potential from the actual practices of self-directed learning. Three of the four measurements of self-directed learning used in this study were measures of students' potential for self-directed learning, not their actual practice. These three measures, self-directed learning readiness, self-directed learning skills, and students' course motivation, were students' own perceptions of themselves rather than actual behavior.

The addition of qualitative measures, including interviews of participants, might reveal motivational components related to students' practice of both problem-based learning and self-directed learning. A qualitative research effort might be useful in better understanding details of the performance component of self-directed learning.

Finally, the attempt to measure self-directed learning performance with time spent on self-directed learning tasks was a good beginning but needs improvement. Future research should address ways to better measure specific aspects of self-directed learning performance.

Conclusions

Students' perceptions of their self-directed learning readiness, self-directed learning skills, and course motivation were not significantly different after experiencing problem-based learning. Problem-based learning may not affect students' perceptions of these self-directed learning components. Possible reasons for observing no differences include: a) the extremely small sample sizes, resulting from high attrition in the course, provided no statistical power, b) the problem-based learning teaching method, while properly conducted, was ineffective due to lack of sufficient student participation, and c) the measures of students' perceptions for self-directed learning readiness, skills, and course motivation may not have been indicative of their actual practices.

The effect of teaching method on self-directed learning performance was significant with the problem-based learning teaching method students having greater self-directed performance. However. questions concerning the accuracy and completeness of the reported data and the small sample size prevent relying too heavily on this conclusion.

While the statistical results indicated significant effect of teaching method. time. and method x time interaction on grades. other factors contributed to the grade differences. Increased difficulty in programming assignments over time coupled with more stringent grading schemes is another possibility. The lower grades of the problem-based group may be a result of less experience with computer programming. a heavier course load. and less available time because of more work hours per week.

More research is needed to measure the quality and level of the problem-based learning teaching method received by students. Measurement methods of self-directed learning performance and behaviors also need to be investigated.

APPENDIX

A. INSTRUCTOR'S GUIDE: PBL EXERCISE 1

Synopsis – Programming Assignment #6 (share with students)

Students are Software Engineers working for the Information Technology department of a large hospital. They are tasked with many different computer related projects including writing software programs that support the hospital's business.

Model Definition of the Problem

The problem is summarized as creating a software program that meets the requirements specified in the scenario.

Subject Matter Objectives

Upon completion of this problem-based learning assignments, students will be able to:

- Understand and use conditional statements in program flow
- Format output
- Read from and write to files
- Understand the importance of the sequence of execution
- Systematically design a program solution
- Suggest, evaluate, and choose from alternative designs
- Understand and apply a systematic approach to problem-solving
- Understand and apply basic software engineering principles of requirements definition, program design, implementation, and testing

The Scenario (share with students)

It's Monday morning and you've got your coffee and are just sitting down at your desk to check your voice mail. You're thinking it's going to be a pleasant week since you're nearly caught up with all your projects and the boss is on vacation for the next two weeks. Working for the IT department for County General Hospital can be really hectic but sometimes you get a break. You might even get to read a little more in that JAVA text you've been studying!

You dial into your voice mail and suddenly that free time you were dreaming about turns into just that – a dream. The boss has a new assignment that must be completed before she gets back from her vacation.

<<See Voice Mail Monolog>>

Well, there went your idea that your boss' vacation would be two extra weeks for your vacation too! Oh well, at least you'll get to learn some of that JAVA you wanted!

The following monolog will be available as an audio recording with the written text version available on request.

Voice Mail Monolog

I hate to drop this project on you and leave on vacation but we got a call from the Accounting Department late Friday. It seems that the docs don't want to do the simple math required for mileage reimbursement on their expense accounts. Every time they travel on hospital business, their mileage is reimbursed on a sliding scale and they don't compute it correctly. While it's not a lot of money, sometimes the Federal Government reimburses the hospital and their bean counters aren't amused by our docs' math skills. In fact, they're threatening to cut off several grants we have that are worth more than $5 million bucks. So you see why our suits upstairs are concerned.

What we have to do is write a simple little program that calculates the reimbursements correctly. I left a copy of the reimbursement scale in your mailbox. Also, our accountants have some pretty specific requirements about formatting the output. I don't know why – maybe it's government regs. You know we're usually at their mercy when they plead government regs so it's got to be just the way they want it or it's not right. I left a sample output with some notes I made when they gave this project to me. I don't really know how this was generated but they said it should do if we fix it up and get it correct. So, your output needs to meet those requirements. You'll see that many of the reimbursement amounts are not correct.

Oh! Did I mention the input data for your program needs to be read from a file and that the output table needs to be written out to different file? This means you'll have one input and one output file – only one of each. They told me that the input file has the number of values to process on the first line of the file – I'm sure we can count on it being an integer. After the first line in the input data file, each line is a single mileage value. Since they are required to report mileage to the nearest tenth, these values must be reals. They couldn't give me an input file; so use the miles on that output sample I left you to create one. You can use it to test the program once you've finished it. I think there were 10 mileage values on that output so the first line of the file will be a ten.

I know you've been reading up on JAVA so why don't you do this program in it. It'll give you a chance to discover some more of its features. I also made a few notes about program structure and such that might help – they're with the other stuff in your mailbox.

Sorry I won't be here to help and unfortunately the cruise ship only allows emergency phone call so you're on your own. You should collaborate with a few other people in the department to see what you're going to need to know to solve the problem. You can also help each other to figure out what resources you might need to use.

Oh! Don't forget to document your program appropriately! Follow the guidelines we've been developing for our programs.

Last thing! Be sure you track your time and activities on the PSP Tracking forms. You know this data will help us better estimate our future projects and keep us funded – besides the fact that it's required!

See you in 2 weeks! I'll expect printed copies of your input datafile, source code, and correctly formatted output file. By the way don't forget to test it!

Have fun! Bye!

Materials to Accompany Problem-Based Learning Scenario – Programming Assignment #6

The following items are to accompany the scenario.

1. Reimbursement Scale
2. Output Sample
3. Programming Notes

Reimbursement scale

Reimbursement scale: round trip mileage	Rate
Up to 500 miles	15 cents per mile
500 to 1000 miles	$ 75.00 plus 12 cents for each mile over 500
1000 to 1500 miles	$135.00 plus 10 cents for each mile over 1000
1500 to 2000 miles	$185.00 plus 8 cents for each mile over 1500
2000 to 3000 miles	$225.00 plus 6 cents for each mile over 2000
over 3000 miles	$285.00 plus 5 cents for each mile over 3000

Output Sample

```
                  MILEAGE REIMBURSMENT

   MILES   REIMBURSEMENT AMOUNT

    40.2      6.03
   799.6    119.94
    -8.6    ***
   499.5    749.25
  5000.2    385.00
  2999.4    334.94
     0.     ***
  1500.1    150.10
  1239.8    154.12
```

Correct Misspellings

Three stars when miles less than or equal 0

Note!
The values in the REIMBURSEMENT AMOUNT Column of this sample are not all correct
(Go figure -- I guess they aren't mathematicians!)
You'll need to get them right!!!!

Line up columns on decimal points –
1 decimal place - accuracy for miles,
2 decimals for amounts

They also want to add totals for reimbursement values, number of mileage values processed, and the number of mileage values that are >= 0.

Be sure you include messages with these! Add them after processing and at bottom of table!

Programming Notes

The following notes should not be given directly to students. When students ask for the programming notes referenced in the scenario, they should be informed that their boss is forgetful and may not have delivered them as promised – a very real and common occurrence in the real world. The instructor should guide the students toward asking appropriate questions in their problem solving process so as to discover these guidelines. It is appropriate that the instructor, as a subject matter resource, conveys this information as student inquiry dictates.

- Use a loop to process your input data on the fly – No need to store any data so do NOT use an array
- Most likely “if/else if” structure will be your best choice for calculating mileage reimbursements > 0
- Besides a main method (that should contain all the variables) you should use at least a separate method to print the heading for the table and another method to output summary information

To use the following classes, include **`import java.text.*`**
NumberFormat class (for formatting output)

For currency – to output total

```
   NumberFormat money =
NumberFormat.getCurrencyInstance();
   System.out.println(money.format(total));
```

Or

DecimalFormat class (also for formatting output)
For one decimal place, rounded – to output total

```
DecimalFormat fmt = new DecimalFormat(“0.0#”);
System.out.println(fmt.format(total));
```

Problem Logs

The problem logs are given to students to guide their problem-based learning activities. Each log constitutes a set of activities corresponding to a step in the problem-based learning process. These logs also serve to aid the instructor's tutoring and facilitation of the process. These logs provide significant scaffolding to guide both the students and instructor. Less scaffolding will be offered as students become more familiar with the problem-based learning activities and as the instructor's tutoring and facilitating skills develop.

Students will be asked to keep an Engineering Notebook, a common practice for many Software Engineers in industry. A notebook will be provided to students to encourage its use and to provide an orderly means of tracking and recording the problem-based learning exercises for these problem logs. Upon completion of the assignment, each student's notebook will be collected as data for research.

Collaborative Groups

The problem logs describe numerous in-class, collaborative group exercises. Collaborative groups of 3-5 students should be created for the duration of the assignment. Note that the collaborative activities center on typical tasks associated with most software development tasks that are done collaboratively in the real world and those reflective activities associated with assessing learning. However, the final program design and implementation is an individual assignment.

Groups may be formed randomly or assigned by the instructor. However, heterogeneous groups can expose students to new ideas and distribute assets and liabilities evenly.

Floating Facilitation

During the collaborative group activities, the instructor follows a floating facilitator model for tutoring and facilitating the group work.

Whole Class Tutoring

The Problem Logs that follow include both in-class and outside of class activities. The outside activities do not require working in collaborative groups while many of the in-class activities are reserved for collaborative group work. In addition to the in-class collaborative activities, designated whole class sessions are devoted to groups sharing their ideas and findings. These sessions are designed to provide opportunities for the instructor to facilitate and guide the overall process of problem-based learning. These sessions should require 10-15 minutes.

Problem Log 1 – What Is This Problem About?
(In-class – Collaborative Group Exercises)

All of the tasks for this Problem Log should be done in class with collaborative groups of 4 to 6 students.

1. **Requirements**
   From what you know so far; list the requirements that you will need to meet to solve this problem. It may be helpful to list them on index cards, one requirement per card. Complete this task with your group by brainstorming for possible requirements. Refine your requirements to include only those necessary to solve the problem. You must complete this task before beginning the next one. *Record your requirements in your Engineering Notebook.*

2. **Beginning Your Investigation of the Problem**
   Before you can implement a solution to this problem, you will need to understand, as fully as possible at this time, the issues and questions posed by each requirement. Carefully think and brainstorm with your group to complete the following task:
   For each requirement, list questions you may have, things you do not understand, and any issues related to that requirement. You may also discover some requirements may cause a need to create other "derived" requirements. Add these derived requirements to your set and analyze each to determine questions and issues. *Record in your Engineering Notebook.*

3. **Thinking About Your Thinking (Metacognition)**
   Because a substantial amount of your employers' money may be lost if you do not create a program that meets your clients' needs, you also have a significant stake in correctly solving this problem. The consequences to you personally are also likely to be significant. What skills and attitudes do you need to bring to this task? *Record in your Engineering Notebook.*

   Often expert problem solvers and competent Software Engineers are asked to do a task that they have never done before or use skills or knowledge they do not currently posses. How do you approach such a situation? *Record in your Engineering Notebook.*

4. **Whole-Class Facilitation**

A list of possible requirements should be boarded by soliciting requirements from each group.

Problem Log 2 – Thinking About Possible Solutions
(In-class – Collaborative Group Exercises)

Collaborate with your group for the following:

1. **Possible solutions**

   It is not necessary at this time that you know how you will implement every detail for you to create designs that will solve the problem. What the program will need to do comes first—how it will do it follows. Vague ideas of how are sufficient at this time.

   Without writing any code, think about what will be needed in your program that will solve this problem. What elements are necessary? What general design might solve this problem? *Record in your Engineering Notebook.*

   What alternatives are there? What would be the advantages and disadvantages of each? Create a simple list with the pros and cons. *Record in your Engineering Notebook.*

   Does each solution meet all the requirements? *Record in your Engineering Notebook.*

2. **Create a list of questions, issues, and unknowns about your possible designs**

   Record in your Engineering Notebook.

Problem Log 3 – Defining What You Know and What You Need To Know

(In-class (1, 2, & 3)-Outside-class (4) – Collaborative Group Exercises (1,2, & 3))

After completing Problem Log 1, you should have a set of requirements annotated with questions, comments, and issues. Problem Log 2 should have given you some ideas about a solution design. Use these to complete the following activities with your collaborative group in class.

1. **Defining Learning Needs (Group Activity)**

   Your next task is to review the requirements and make a list of what you already know that applies to this problem and a second list of what you think you will need to learn. Next, use your possible designs to add more items of what you will need to learn to each list you just created. After creating these two lists for your group, make copies for each individual. Later outside of class, you (individually) should repeat this exercise to better understand your specific learning needs. *Record in your Engineering Notebook.*

2. **Defining Possible Resources (Group Activity)**

   For the learning needs you defined above, what sources of information should you use to acquire the knowledge and skills? It may be helpful to review your list of needs to determine appropriate resources. *Record in your Engineering Notebook.*

3. **Thinking About Your Thinking (Metacognition)**

   How difficult was it to determine what you did not know? Were there things you thought you understood or knew that, upon reflection, either individually or with your group, you realized you needed to learn? *Record in your Engineering Notebook.*

   Which items in your list of things to learn are likely to require your instructor as a resource?

   Can your peers be a resource?

   How well are you able to follow this process? Why?

4. **Repeat Steps 1, 2, and 3 (Individually – Outside of Class)**

   Personalize the list of learning needs and the list of things known for your self. *Record in your Engineering Notebook.*

Problem Log 4 – Using Your Resources

(Outside-class – Individual Exercises)

1. **Use your list of resources to satisfy your learning needs**
   Use the appropriate resources to develop the understanding, knowledge, and skills you have defined for yourself for each need.

2. **Summarizing What You Learned**
   In your notes, briefly summarize what you learned for each need. *Record in your Engineering Notebook.*

3. **Thinking About Your Thinking (Metacognition)**
   What worked well in this process? What did not? Why? What should you do differently next time? *Record in your Engineering Notebook.*

Problem Log 5 – Reviewing What You Learned

(In-class – Collaborative Group Exercises)

With your group, use your lists of learning needs to share what you have learned with other members of your group. Relate what you learned to what you needed to learn.

1. **Reviewing Learning**

   Briefly, answer the following questions as they apply to your experience.

   What interesting or valuable information did others learn that is valuable to solving the problem?

   How does your new knowledge change your view of the problem?

   What, if anything learned by others, causes you to revise what you have learned?

2. **Reviewing How You Learned**

   Briefly, answer the following questions as they apply to your experience.

   What **resources** for learning did others use that is valuable to solving the problem?

   What **strategies** for learning did others use that is valuable to solving the problem?

   What **resources and strategies**, if any, were of little value toward solving the problem?

Problem Log 6 – Designing a Solution

(Outside-class – Individual Exercises)

It's time to define an appropriate solution to the problem.

1. **Selecting a Solution**

   Before writing code, think about the possible solutions previously discussed in class and with your group. What are the advantages and disadvantages previously discussed? Are you able to decide on a design that meets requirements and can be implemented with what you have learned?

2. **Program Design**

   After deciding upon a design, document it in your Engineering Notebook.

Problem Log 7 – Completing the Program

(Outside-class – Individual Exercise)

1. **Code**

   Using your design, code your solution!

2. **Thinking About Your Thinking (Metacognition)**

   Did your newly acquired knowledge and skills make your programming task easier? Was it more efficient? Did you find you needed to learn more during the coding phase?

Problem Log 8 – Testing the Solution

(Outside-class – Individual Exercise)

1. **Test Design**

   Software engineers must test each individual part of the code they write during Unit Testing. While this is often less-well formalized, the completed program should be more formally tested. Using your list of requirements, design a set of test cases that will test each requirement to be sure it is satisfied (validation). These tests should also evaluate whether the program does things correctly and defect free (verification). *Record your test cases in your Engineering Notebook.*

2. **Test Execution**

   Run your test cases with your program. *Record your results in your Engineering Notebook.*

Problem Log 9 – Final Review
(In-class – Collaborative Group Exercises)

1. **Summarizing What You Learned**

   Work with your group to summarize the most important things you learned during this assignment.

   What programming principles did you learn? What did you learn about your ability to program? What did you learn about your ability to approach a real-world problem?

   Brainstorm to create a list and *record in your Engineering Notebook.*

2. **Thinking About Your Thinking (Metacognition)**

   Review with your group the processes and activities you experienced for this assignment. Create an outline or model about the process that might be useful for future learning. *Record in your Engineering Notebook.*

B. NON-PBL EXERCISE 1

The Mathematical Association of America hosts an annual summer meeting. Each state sends one official delegate to the section officers' meeting at this summer session. The national organization reimburses the official state delegates according to the scale at the bottom of the page. Write a Java program that will calculate the reimbursement values for the data at the bottom of the page. Your program must satisfy the following:

1. It reads data from a file and writes data to a file – there should be only one output file and only one input file.
2. The first line of the data file will contain the number (an integer) of data values to process. After the first line of the data file, each line will contain a real number representing the number of miles. Use a for loop to process the mileage values. **Do NOT use an array.**
3. Use an "if/else if" statement and the scale below to calculate the mileage reimbursement if the input value is > 0.
4. The main method should contain all the variables – do not use a separate class for storage. Use at least the following methods:
   a. Method to print heading for the table
   b. Method to output summary information
5. There should be one line of output for each mileage value processed – use a table form for the output. The table should be lined up by the decimal point. Each detail line of the table will contain the number of miles (real – print with 1 decimal place) and the reimbursement amount (real – print with 2 decimal places). If the input value is <= 0, then output 3 stars in place of the reimbursement amount.
6. After all the data values have been processed, print the total of the reimbursement values, the number of mileage values processed, and the number of mileage values that were >= 0, include messages.

You will need to create a data file containing the following data – the file should contain one number per line.

10 40.2 799.6 -8.6 499.5 5000.2 5000 2999.4 0 1500.1 1239.8

Reimbursement scale:

Round trip mileage	rate
up to 500 miles	15 cents per mile
500 to 1000 miles	$ 75.00 plus 12 cents for each mile over 500
1000 to 1500 miles	$135.00 plus 10 cents for each mile over 1000
1500 to 2000 miles	$185.00 plus 8 cents for each mile over 1500
2000 to 3000 miles	$225.00 plus 6 cents for each mile over 2000
over 3000 miles	$285.00 plus 5 cents for each mile over 3000

Be sure to appropriately document this program – use guidelines handed out.
Hand in printed copies of your data file, the source code, and the output.

To use the classes below, include import java.text.*
Formatting Output – the NumberFormat class can be used
for currency – to output total

```
NumberFormat money = NumberFormat.getCurrencyInstance():
```

```
          System.out.println(money.format(total));
or DecimalFormat class can be used
      for one decimal place, rounded – to output total
          DecimalFormat fmt = new DecimalFormat("0.0#");
          System.out.println(fmt.format(total));
```

C. INSTRUCTOR'S GUIDE: PBL EXERCISE 2

Synopsis – Programming Assignment #7 (share with students)

Students are Software Engineers working for the Computer Infrastructure Support Group within the Information Technologies Department of a large State University. This group is responsible for custom software development assignments requested by the administration of the university.

Model Definition of the Problem

The problem is summarized as creating a software program that meets the requirements specified in the scenario.

Subject Matter Objectives

Upon completion of this problem-based learning assignments, students will be able to:

- Understand and use conditional statements in program flow
- Read and write String data from and to a file
- Format output
- Read from and write to files
- Process files of unknown length
- Understand the importance of the sequence of execution
- Systematically design a program solution
- Suggest, evaluate, and choose from alternative designs
- Understand and apply a systematic approach to problem-solving
- Understand and apply basic software engineering principles of requirements definition, program design, implementation, and testing

The Scenario (share with students)

You are a Software Engineer working for the Computer Infrastructure Support Group (CISG) within the Information Technologies Department of a large State University. Your group is responsible for custom software development assignments requested by the administration of the university.

In yesterday's email there was an invitation to a Friday afternoon party to celebrate the successful delivery of your last project (ADMISS—EVAL). Just last week your group completed the new admissions evaluation software that creates a composite score for each new student applicant. The Dean of Admissions is so pleased with the system your department delivered she wants to celebrate. This is unusual!

That was a tough project. There were so many factors to evaluate to determine if a potential student should be admitted. The factors were weighted differently and had many exceptions and special considerations. However, by closely working with Bob in the admissions office, you were able to create a useful (and successful) algorithm. Each student has an admission score computed from all the available factors for that student. These input factors included things like SAT scores, ACT scores, high school GPA, previous college GPA, entrance essay scores, interview ratings, and the university's weighting factors that are assigned for each component. The algorithm also incorporates other student factors such as in-state resident and non-resident since there is a mandate to mix and balance the student population for each semester. Admissions can then use the computed score to accept or reject an applicant.

Unfortunately, the celebration is clouded by the next email in your inbox. It's from Bob, the admissions technical project client manager. It seems that during the intense work that your team and the admissions office personnel did to develop the algorithm, one minor requirement was overlooked. An output file is needed that can be forwarded to the State Higher Education Commission for some later statistical analysis. You print out the email since it has a fairly complete set of requirements that will satisfy the State Higher Education Commission. (See Attached email).

Since this task is not too large and the project was such a success, you don't want to risk the project by incorporating this new task into system. You decide that a small stand-alone program that will use an intermediate output file from the system is the best approach. Besides, this will likely change next semester because the politicians are always meddling in the Higher Education Commission's business and they have kept you in new projects since you came to work here!

Materials to Accompany Problem-Based Learning Scenario – Programming Assignment #6

The following items are to accompany the scenario.

1. First Email – Invitation
2. Second Email – URGENT Task
3. File data for input to new program

First Email

From: Sue Smith <smiths@state.edu>
To: CISG, ADMISS-Eval Team
Subject: Thanks! Congratulations!

Congratulations on successfully completing the ADMISS-EVAL project! As Dean of Admissions, I'd like to thank you for your efforts and accomplishments. To show our appreciation, you are invited to a pizza party next Friday. Please let me know if you will be able to attend.

Thanks again!

Sue Smith
Dean of Admissions

Second Email

From: Bob Johnson <johnsonw@state.edu>
To: CISG, ADMISS-Eval Team
Subject: URGENT – One More Task

I hate to be the bearer of bad news but we have another task to add that is high priority and critical. We forgot one other output that we must have from the ADMISS-EVAL system. We need an output file that can be sent to the State Higher Education Commission that contains some specific data they want concerning our admissions decisions on each student.

I've tried to outline some of the specific requirements but if anything is missing be sure to ask.

1. They need a printed report (written to a file) that consists of a table with appropriate headings.

2. The columns of the report must contain the name of the student, value (composite score for each new student applicant), and then a message.
3. The message is SCHOLARSHIP if the value is 90 or more; Admit if the value is between 70 and 89 inclusive; REJECT otherwise.

After all the data lines in the file have been processed (I'm assuming you will get an input data file from our new system's output file), print with messages the number of data lines processed, and the number and real average of the values between 70 and 89 inclusive (use 1 decimal place).

I don't really know what the State Higher Education Commission wants this for but you know how political it can be. We'll just have to give them what they want until they change their minds again!

BTW, Be sure you track your time and activities on that PSP Tracking form I saw you using. I think we can get reimbursed for your time from the State Higher Education Commission.

When you get done with the program and testing it, send me a printed copy of the source code, the data file, and the output so I can look at it before we send it on.

Thanks,
Bob

Output File to Use for Input to Your New Program

Notes: the admission score in the first column is an integer and the name is a string; the file size may be of any length (variable number of records).

```
40             Light Karen L
81             Fagan Bert Todd
60             Antrim Forrest N
95             Camden Warren
52             Mulicka Al B
89             Lee Phoebe
75             Bright Harry
92             Garris Ted
43             Benson Martyne
99             Lloyd Jeanine D
73             Leslie Bennie A
70             Brandt Leslie
39             Schulman David
90             Worthington Dan
76             Hall Gus W
50             Prigeon Dale P
63             Fitzgibbons Rusty
```

Programming Notes

The following notes should not be given directly to students. The instructor should guide the students toward asking appropriate questions in their problem solving process so as to discover these guidelines. It is appropriate that the instructor, as a subject matter resource, conveys this information as student inquiry dictates.

- Use a loop to process your input data on the fly – No need to store any data so do NOT use an array
- Most likely "if/else if" structure will be your best choice for processing
- Methods should be used for the heading of the table and for the summary of the table

A sample of source code for processing a line from the input file follows:

```
// reads one line of user input where the input has the form
//     integer       name
// name may consist of any number of parts
// output is a message to the screen with the integer and the name

import java.io.*;
import java.util.*;

class intName
{
  public static void main(String [] args) throws IOException
  { System.out.println("This is a new example.");
    int num;
    String name;
    BufferedReader br = new Buffered Peader (new
    InputStreamReader(System.in));
    String str = br.readLine();
    StringTokenizer st = new StringTokenizer(str);
    num = Integer.parseInt(st.nextToken());
    name = st.nextToken();
    while (st.hasMoreTokens())
        { name = name + ' ' + st.nextToken();}
    System.out.println("number is " + num);
    System.out.println("name is " + name);
  }
}
```

Problem Logs

The problem logs are given to students to guide their problem-based learning activities. Each log constitutes a set of activities corresponding to a step in the problem-based learning process. These logs also serve to aid the instructor's tutoring and facilitation of the process. These logs provide significant scaffolding to guide both the students and instructor. Less scaffolding will be offered as students become more familiar with the problem-based learning activities and as the instructor's tutoring and facilitating skills develop.

Students will be asked to keep an Engineering Notebook. a common practice for many Software Engineers in industry. A notebook will be provided to students to encourage its use and to provide an orderly means of tracking and recording the problem-based learning exercises for these problem logs. Upon completion of the assignment, each student's notebook will be collected as data for research.

Collaborative Groups

The problem logs describe numerous in-class, collaborative group exercises. Collaborative groups of 3-5 students should be created for the duration of the assignment. Note that the collaborative activities center on typical tasks associated with most software development tasks that are done collaboratively in the real world and those reflective activities associated with assessing learning. However. the final program design and implementation is an individual assignment.

Groups may be formed randomly or assigned by the instructor. However, heterogeneous groups can expose students to new ideas and distribute assets and liabilities evenly.

Floating Facilitation

During the collaborative group activities, the instructor follows a floating facilitator model for tutoring and facilitating the group work.

Whole Class Tutoring

The Problem Logs that follow include both in-class and outside of class activities. The outside activities do not require working in collaborative groups while many of the in-class activities are reserved for collaborative group work. In addition to the in-class collaborative activities, designated whole class sessions are devoted to groups sharing their ideas and findings. These sessions are designed to provide opportunities for the instructor to facilitate and guide the overall process of problem-based learning. These sessions should require 10-15 minutes.

Problem Log 1 – What Is This Problem About?

(In-class – Collaborative Group Exercises)

All of the tasks for this Problem Log should be done in class with collaborative groups of 4 to 6 students.

1. **Requirements**

   From what you know so far; list the requirements that you will need to meet to solve this problem. It may be helpful to list them on index cards, one requirement per card. Complete this task with your group by brainstorming for possible requirements. Refine your requirements to include only those necessary to solve the problem. You must complete this task before beginning the next one. *Record your requirements in your Engineering Notebook.*

2. **Beginning Your Investigation of the Problem**

   Before you can implement a solution to this problem, you will need to understand, as fully as possible at this time, the issues and questions posed by each requirement. Carefully think and brainstorm with your group to complete the following task:

   For each requirement, list questions you may have, things you do not understand, and any issues related to that requirement. You may also discover some requirements may cause a need to create other "derived" requirements. Add these derived requirements to your set and analyze each to determine questions and issues. *Record in your Engineering Notebook.*

3. **Thinking About Your Thinking (Metacognition)**

   Because a substantial amount of your employers' money may be lost if you do not create a program that meets your clients' needs, you also have a significant stake in correctly solving this problem. The consequences to you personally are also likely to be significant. What skills and attitudes do you need to bring to this task? *Record in your Engineering Notebook.*

   Often expert problem solvers and competent Software Engineers are asked to do a task that they have never done before or use skills or knowledge they do not currently posses. How do you approach such a situation? *Record in your Engineering Notebook.*

4. **Whole-Class Facilitation**

A list of possible requirements should be boarded by soliciting requirements from each group.

Problem Log 2 – Thinking About Possible Solutions
(In-class – Collaborative Group Exercises)

Collaborate with your group for the following:

1. **Possible solutions**

   It is not necessary at this time that you know how you will implement every detail for you to create designs that will solve the problem. What the program will need to do comes first—how it will do it follows. Vague ideas of how are sufficient at this time.

   Without writing any code. think about what will be needed in your program that will solve this problem. What elements are necessary? What general design might solve this problem? *Record in your Engineering Notebook.*

   What alternatives are there? What would be the advantages and disadvantages of each? Create a simple list with the pros and cons. *Record in your Engineering Notebook.*

   Does each solution meet all the requirements? *Record in your Engineering Notebook.*

2. **Create a list of questions, issues. and unknowns about your possible designs**

   Record in your Engineering Notebook.

Problem Log 3 – Defining What You Know and What You Need To Know

(In-class (1, 2, & 3)-Outside-class (4) – Collaborative Group Exercises (1,2, & 3))

After completing Problem Log 1, you should have a set of requirements annotated with questions, comments, and issues. Problem Log 2 should have given you some ideas about a solution design. Use these to complete the following activities with your collaborative group in class.

1. **Defining Learning Needs (Group Activity)**
   Your next task is to review the requirements and make a list of what you already know that applies to this problem and a second list of what you think you will need to learn. Next, use your possible designs to add more items of what you will need to learn to each list you just created. After creating these two lists for your group, make copies for each individual. Later outside of class, you (individually) should repeat this exercise to better understand your specific learning needs. *Record in your Engineering Notebook.*

2. **Defining Possible Resources (Group Activity)**
   For the learning needs you defined above, what sources of information should you use to acquire the knowledge and skills? It may be helpful to review your list of needs to determine appropriate resources. *Record in your Engineering Notebook.*

3. **Thinking About Your Thinking (Metacognition)**
   How difficult was it to determine what you did not know? Were there things you thought you understood or knew that, upon reflection, either individually or with your group, you realized you needed to learn? *Record in your Engineering Notebook.*

   Which items in your list of things to learn are likely to require your instructor as a resource?

   Can your peers be a resource?

   How well are you able to follow this process? Why?

4. **Repeat Steps 1, 2, and 3 (Individually – Outside of Class)**
   Personalize the list of learning needs and the list of things known for your self. *Record in your Engineering Notebook.*

Problem Log 4 – Using Your Resources

(Outside-class – Individual Exercises)

1. **Use your list of resources to satisfy your learning needs**
   Use the appropriate resources to develop the understanding, knowledge, and skills you have defined for yourself for each need.

2. **Summarizing What You Learned**
   In your notes, briefly summarize what you learned for each need. *Record in your Engineering Notebook.*

3. **Thinking About Your Thinking (Metacognition)**
   What worked well in this process? What did not? Why? What should you do differently next time? *Record in your Engineering Notebook.*

Problem Log 5 – Reviewing What You Learned
(In-class – Collaborative Group Exercises)

With your group, use your lists of learning needs to share what you have learned with other members of your group. Relate what you learned to what you needed to learn.

1. **Reviewing Learning**
   Briefly, answer the following questions as they apply to your experience.

   What interesting or valuable information did others learn that is valuable to solving the problem?

   How does your new knowledge change your view of the problem?

   What, if anything learned by others, causes you to revise what you have learned?

2. **Reviewing How You Learned**
   Briefly, answer the following questions as they apply to your experience.

   What **resources** for learning did others use that is valuable to solving the problem?

   What **strategies** for learning did others use that is valuable to solving the problem?

   What **resources and strategies**, if any, were of little value toward solving the problem?

Problem Log 6 – Designing a Solution

(Outside-class – Individual Exercises)

It's time to define an appropriate solution to the problem.

1. **Selecting a Solution**

   Before writing code, think about the possible solutions previously discussed in class and with your group. What are the advantages and disadvantages previously discussed? Are you able to decide on a design that meets requirements and can be implemented with what you have learned?

2. **Program Design**

   After deciding upon a design, document it in your Engineering Notebook.

Problem Log 7 – Completing the Program
(Outside-class – Individual Exercise)

1. **Code**

   Using your design, code your solution!

2. **Thinking About Your Thinking (Metacognition)**

   Did your newly acquired knowledge and skills make your programming task easier? Was it more efficient? Did you find you needed to learn more during the coding phase?

Problem Log 8 – Testing the Solution

(Outside-class – Individual Exercise)

1. **Test Design**

   Software engineers must test each individual part of the code they write during Unit Testing. While this is often less-well formalized, the completed program should be more formally tested. Using your list of requirements, design a set of test cases that will test each requirement to be sure it is satisfied (validation). These tests should also evaluate whether the program does things correctly and defect free (verification). *Record your test cases in your Engineering Notebook.*

2. **Test Execution**

   Run your test cases with your program. *Record your results in your Engineering Notebook.*

Problem Log 9 – Final Review
(In-class – Collaborative Group Exercises)

1. **Summarizing What You Learned**

   Work with your group to summarize the most important things you learned during this assignment.

   What programming principles did you learn? What did you learn about your ability to program? What did you learn about your ability to approach a real-world problem?

   Brainstorm to create a list and *record in your Engineering Notebook.*

2. **Thinking About Your Thinking (Metacognition)**

   Review with your group the processes and activities you experienced for this assignment. Create an outline or model about the process that might be useful for future learning. *Record in your Engineering Notebook.*

D. NON-PBL EXERCISE 2

Write a Java program that reads from a file and writes to a file. It will process data lines until end of file. The data in the input file should be read and processed in one pass thru the file. All output should be to one file. Each line in the data file will contain an integer followed by a name. Use a String to store the name. See example at the bottom of the page for a sample program which inputs one such line. **Do NOT use an array to store the integers.**

Create a printed report that consists of a table with appropriate headings. The columns of the report will contain the name, value, and then a message. The message is OUTSTANDING if the value is 90 or more; Satisfactory if the value is between 70 and 89 inclusive; FAILING otherwise. Use an "if/else if".

After all the data lines have been processed, print with messages the number of data lines processed, and the number and real average of the values between 70 and 89 inclusive (use 1 decimal place). Follow the documentation guidelines. Methods must be used for the heading of the table and for the summary of the table. Hand in a printed copy of the source code, the data file, and the output.

```
Data to use:       Use order here:
------------------------------------------------------
  40               Light Karen L
  81               Fagan Bert Todd
  60               Antrim Forrest N
  95               Camden Warren
  52               Mulicka Al B
  89               Lee Phoebe
  75               Bright Harry
  92               Garris Ted
  43               Benson Martyne
  99               Lloyd Jeanine D
  73               Leslie Bennie A
  70               Brandt Leslie
  89               Schulman David
  90               Worthington Dan
  70               Hall Gus W
  50               Prigeon Dale R
  63               Fitzgibbons Rusty
------------------------------------------------------
```

```
// reads one line of user input where the input has the form
//     integer      name
// name may consist of any number of parts
// output is a message to the screen with the integer and the name
```

```
import java.io.*;
import java.util.*;

class intName
{
  public static void main(String [] args) throws IOException
  { System.out.println("This is a new example.");
    int num;
    String name;
    BufferedReader br = new Buffered Reader (new
    InputStreamReader(System.in));
    String str = br.readLine();
    StringTokenizer st = new StringTokenizer(str);
    num = Integer.parseInt(st.nextToken());
    name = st.nextToken();
    while (st.hasMoreTokens())
        {  name = name + ' ' + st.nextToken();}
    System.out.println("number is " + num);
    System.out.println("name is " + name);
  }
}
```

E. DEMOGRAPHICS SURVEY

DEMOGRAPHIC INFORMATION

1. Gender (circle one).   Male   Female

2. Birthdate ____________________

3. What year did you graduate from high school?

   ____

4. Class level (circle one).

   Freshman   Sophomore   Junior   Senior

5. Ethnic background (circle one).

   Afro-American   Asian   Caucasian   Hispanic
   Other
   or Black   American   or Spanish
   Speaking

6. How many hours per week do you work for pay?

   ____

7. How many other college level courses have you
   had in this subject area?

   ____

8. How many classes are you taking this term?

   ____

9. How many hours a week do you study for this course?

   ____

10. Reason for taking this class (circle yes or no for each item).

a.	fulfills distribution requirement	Yes	No
b.	content seems interesting	Yes	No
c.	is required of all students at college	Yes	No
d.	will be useful to me in other courses	Yes	No
e.	is an easy elective	Yes	No
f.	will help improve my academic skills	Yes	No
g.	is required for major (program)	Yes	No
h.	was recommended by a friend	Yes	No
i.	was recommended by a counselor	Yes	No
j.	will improve career prospects	Yes	No
k.	fits into my schedule	Yes	No

F. SDLRS QUESTIONNAIRE

QUESTIONNAIRE[1]

INSTRUCTIONS: This is a questionnaire designed to gather data on learning preferences and attitudes towards learning. After reading each item, please indicate the degree to which you feel that statement is true of you. Please read each choice carefully and circle the number of the response which best expresses your feeling.

There is no time limit for the questionnaire. Try not to spend too much time on any one item, however. Your first reaction to the question will usually be the most accurate.

		RESPONSES				
	ITEMS:	Almost never true of me; I hardly ever feel this way.	Not often true of me; I feel this way less than half the time.	Sometimes true of me; I feel this way about half the time.	Usually true of me; I feel this way more than half the time.	Almost always true of me; there are very few times when I don't feel this
1.	I'm looking forward to learning as long as I'm living.	1	2	3	4	5
2.	I know what I want to learn.	1	2	3	4	5
3.	When I see something that I don't understand, I stay away from it.	1	2	3	4	5
4.	If there is something I want to learn, I can figure out a way to learn it.	1	2	3	4	5
5.	I love to learn.	1	2	3	4	5
6.	It takes me a while to get started on new projects.	1	2	3	4	5

	RESPONSES				
ITEMS:	Almost never true of me; I hardly ever feel this way.	Not often true of me; I feel this way less than half the time.	Sometimes true of me; I feel this way about half the time.	Usually true of me; I feel this way more than half the time.	Almost always true of me; there are very few times when I don't feel this
7. In a classroom, I expect the teacher to tell all class members exactly what to do at all times.	1	2	3	4	5
8. I believe that thinking about who you are, where you are, and where you are going should be a major part of every person's education.	1	2	3	4	5
9. I don't work very well on my own.	1	2	3	4	5
10. If I discover a need for information that I don't have, I know where to go to get it.	1	2	3	4	5
11. I can learn things on my own better than most people.	1	2	3	4	5
12. Even if I have a great idea, I can't seem to develop a plan for making it work.	1	2	3	4	5
13. In a learning experience, I prefer to take part in deciding what will be learned and how.	1	2	3	4	5
14. Difficult study doesn't bother me if I'm interested in something.	1	2	3	4	5
15. No one but me is truly responsible for what I learn.	1	2	3	4	5
16. I can tell whether I'm learning something well or not.	1	2	3	4	5

	RESPONSES				
ITEMS:	Almost never true of me; I hardly ever feel this way.	Not often true of me; I feel this way less than half the time.	Sometimes true of me; I feel this way about half the time.	Usually true of me; I feel this way more than half the time.	Almost always true of me; there are very few times when I don't feel this
17. There are so many things I want to learn that I wish that there were more hours in a day.	1	2	3	4	5
18. If there is something I have decided to learn, I can find time for it, no matter how busy I am.	1	2	3	4	5
19. Understanding what I read is a problem for me.	1	2	3	4	5
20. If I don't learn, it's not my fault.	1	2	3	4	5
21. I know when I need to learn more about something.	1	2	3	4	5
22. If I can understand something well enough to get a good grade on a test, it doesn't bother me if I still have questions about it.	1	2	3	4	5
23. I think libraries are boring places.	1	2	3	4	5
24. The people I admire most are always learning new things.	1	2	3	4	5
25. I can think of many different ways to learn about a new topic.	1	2	3	4	5
26. I try to relate what I am learning to my long-term goals.	1	2	3	4	5
27. I am capable of learning for myself almost anything I might need to know.	1	2	3	4	5

	RESPONSES				
ITEMS:	Almost never true of me; I hardly ever feel this way.	Not often true of me; I feel this way less than half the time.	Sometimes true of me; I feel this way about half the time.	Usually true of me; I feel this way more than half the time.	Almost always true of me; there are very few times when I don't feel this
28. I really enjoy tracking down the answer to a question.	1	2	3	4	5
29. I don't like dealing with questions where there is not one right answer.	1	2	3	4	5
30. I have a lot of curiosity about things.	1	2	3	4	5
31. I'll be glad when I'm finished learning.	1	2	3	4	5
32. I'm not as interested in learning as some other people seem to be.	1	2	3	4	5
33. I don't have any problems with basic study skills.	1	2	3	4	5
34. I like to try new things, even if I'm not sure how they will turn out.	1	2	3	4	5
35. I don't like it when people who really know what they're doing point out mistakes that I am making.	1	2	3	4	5
36. I'm good at thinking of unusual ways to do things.	1	2	3	4	5
37. I like to think about the future.	1	2	3	4	5
38. I'm better than most people are at trying to find out the things I need to know.	1	2	3	4	5
39. I think of problems as challenges, not stopsigns.	1	2	3	4	5

	RESPONSES				
ITEMS:	Almost never true of me; I hardly ever feel this way.	Not often true of me; I feel this way less than half the time.	Sometimes true of me; I feel this way about half the time.	Usually true of me; I feel this way more than half the time.	Almost always true of me; there are very few times when I don't feel this
40. I can make myself do what I think I should.	1	2	3	4	5
41. I'm happy with the way I investigate problems.	1	2	3	4	5
42. I become a leader in group learning situations.	1	2	3	4	5
43. I enjoy discussing ideas.	1	2	3	4	5
44. I don't like challenging learning situations.	1	2	3	4	5
45. I have a strong desire to learn new things.	1	2	3	4	5
46. The more I learn, the more exciting the world becomes.	1	2	3	4	5
47. Learning is fun.	1	2	3	4	5
48. It's better to stick with the learning methods that we know will work instead of always trying new ones.	1	2	3	4	5
49. I want to learn more so that I can keep growing as a person.	1	2	3	4	5
50. I am responsible for my learning – no one else is.	1	2	3	4	5
51. Learning how to learn is important to me.	1	2	3	4	5
52. I will never be too old to learn new things.	1	2	3	4	5

	RESPONSES				
ITEMS:	Almost never true of me; I hardly ever feel this way.	Not often true of me; I feel this way less than half the time.	Sometimes true of me; I feel this way about *half the time.*	Usually true of me; I feel this way more than half the time.	Almost always true of me; there are very few times when I don't feel this
53. Constant learning is a bore.	1	2	3	4	5
54. Learning is a tool for life.	1	2	3	4	5
55. I learn several new things on my own each year.	1	2	3	4	5
56. Learning doesn't make any difference in my life.	1	2	3	4	5
57. I am an effective learner in the classroom and on my own.	1	2	3	4	5
58. Learners are leaders.	1	2	3	4	5

[1] Copyright 1977 Lucy M. Guglielmino, used with permission

G. MSLQ QUESTIONNAIRE

Motivated Strategies For Learning Questionnaire[2]

Part A. Motivation

The following questions ask about your motivation for and attitudes about this class. **Remember there are no right or wrong answers, just 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 the 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.	In a class like this, I prefer course material that really challenges me so I can learn new things.	1	2	3	4	5	6	7
2.	If I study in appropriate ways, then I will be able to learn the material in this course.	1	2	3	4	5	6	7
3.	When I take a test I think about how poorly I am doing compared with other students.	1	2	3	4	5	6	7
4.	I think I will be able to use what I learn in this course in other courses.	1	2	3	4	5	6	7

		not at all true of me						**very true of me**
5.	I believe I will receive an excellent grade in this class.	1	2	3	4	5	6	7
6.	I'm certain I can understand the most difficult material presented in the reading for this course.	1	2	3	4	5	6	7
7.	Getting a good grade in this class is the most satisfying thing for me right now.	1	2	3	4	5	6	7
8.	When I take a test I think about items on other parts of the test I can't answer.	1	2	3	4	5	6	7
9.	It is my own fault if I don't learn the material in this course.	1	2	3	4	5	6	7
10.	It is important for me to learn the course material in this class.	1	2	3	4	5	6	7
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

		not at all true of me						very true of me
12.	I'm confident I can learn the basic concepts taught in this class.	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 test 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

		not at all true of me						very true of me
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
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

		not at all true of me						very true of me
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

Part B. Learning Strategies

The following questions ask about your learning strategies and study skills for this class. Again, there are no right or wrong answers. Answer the questions about how you study in this class 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 the 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**

32. When I study the readings 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

not at all true of me — very true of me

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

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

not at all true of me **very true of me**

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 in the readings. I try to decide if there is good supporting evidence. 1 2 3 4 5 6 7

not at all true of me — very true of me

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. 1 2 3 4 5 6 7

53. When I study for this class, I pull together information from different sources, such as lectures, readings, and discussions. 1 2 3 4 5 6 7

not at all true of me **very true of me**

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

not at all true of me **very true of me**

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 related the material to what I already know. 1 2 3 4 5 6 7

65. I have a regular place set aside for studying. 1 2 3 4 5 6 7

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

not at all true of me **very true of me**

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 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

		not at all true of me			very true of me			
73.	I attend this class regularly.	1	2	3	4	5	6	7
74.	Even when course materials are dull and uninteresting, I mange 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
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

		not at all true of me			very true of me			
80.	I rarely find time to review my notes or 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.	1	2	3	4	5	6	7

[2] Copyright 1991 The Regents of The University of Michigan, used with permission

H. PSP TIME LOGS

ID Code: ____________________ **Date:** ____________________

Date	Start	Stop	Interruption Time	Delta Time	Activity1	Comments

Activity Codes1 (See reverse for full description)

(RD)	– Requirements Definition
(LN)	– Defining Learning Needs
(LR)	– Defining Learning Resources
(D)	– Designing Solutions
(L/S)	– Learning/Study
(P)	– Programming/Implementation
(DB)	– Debugging
(T)	– Testing
(RL)	– Reviewing Learning

Activity Codes[1]

Requirements Definition (RD) – Working on understanding & defining what's needed to solve the problem; analyzing the problem

Defining Learning Needs (LN) – Thinking & understanding what you will need to know to solve the problem

Defining Learning Resources (LR) – Thinking & finding the resources you will use to learn what you need to solve the problem

Designing Solutions (D) – Thinking & creating possible solutions to each requirement you have identified (prior to programming)

Learning/Study (L/S) – Time spent reading, understanding, *collaborating/sharing information*, and experimenting for the purpose of learning what you need to solve the problem

Programming/Implementation (P) – Actual writing code to implement the solution you have designed; does NOT include experimentation to find a solution to a particular issue (this is Learning/Study)

Debugging (DB) – Finding & fixing coding errors & errors in logic

Testing (T) – Checking your program for conformance to requirements and correctness; searching for defects

Reviewing Learning (RL) – Thinking about and reviewing what you learned, what you wanted or needed to learn for this problem, and how well you were able to define and use learning resources

PSP Weekly Summary

ID Code: ____________________ **Date:** ____________________

Week Starting ______________

TASKS → Day ↓	Reqmts Define (RD)	Define Learn Needs (LN)	Define Learning Resource (LR)	Design Solution (D)	Learning /Study (L/S)	Prog (P)	Debug (DB)	Testing (T)	Review Learn (RL)	Totals
Sun										
Mon										
Tue										
Wed										
Thur										
Fri										
Sat										
TOTALS										

Activity Codes[1] (See reverse for full description)

(RD) – Requirements Definition
(LN) – Defining Learning Needs
(LR) – Defining Learning Resources
(D) – Designing Solutions
(L/S) – Learning/Study
(P) – Programming/Implementation
(DB) – Debugging
(T) – Testing
(RL) – Reviewing Learning

Activity Codes[1]

Requirements Definition (RD) – Working on understanding & defining what's needed to solve the problem; analyzing the problem

Defining Learning Needs (LN) – Thinking & understanding what you will need to know to solve the problem

Defining Learning Resources (LR) – Thinking & finding the resources you will use to learn what you need to solve the problem

Designing Solutions (D) – Thinking & creating possible solutions to each requirement you have identified (prior to programming)

Learning/Study (L/S) – Time spent reading, understanding, collaborating/sharing information, and experimenting for the purpose of learning what you need to solve the problem

Programming/Implementation (P) – Actual writing code to implement the solution you have designed; does NOT include experimentation to find a solution to a particular issue (this is Learning/Study)

Debugging (DB) – Finding & fixing coding errors & errors in logic

Testing (T) – Checking your program for conformance to requirements and correctness; searching for defects

Reviewing Learning (RL) – Thinking about and reviewing what you learned, what you wanted or needed to learn for this problem, and how well you were able to define and use learning resources

I. REPEATED MEASURES ANOVA ASSUMPTIONS

Levene's Test of Equality of Error Variances[a]

	F	df1	df2	Sig.
MSLQA1	1.596	1	14	.227
MSLQA2	.619	1	14	.444
MSLQA3	.321	1	14	.580
MSLQB1	8.721	1	14	.010
MSLQB2	.882	1	14	.363
MSLQB3	.007	1	14	.935
SDLRS1	.010	1	14	.923
SDLRS2	.051	1	14	.825
SDLRS3	.591	1	14	.455
GRADE15	5.785	1	14	.031
PRG6PCT	.836	1	14	.376
PRG7PCT	1.953	1	14	.184
SDL2	.151	1	10	.706
SDL2	.091	1	10	.769

Tests the null hypothesis that the error variance of the dependent variable is equal across groups.

a Design: Intercept+SECTION Within Subjects Design: TIME

Mauchly's Test of Sphericityb

		Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[a]		
Within Subjects Effect	Measure					Greenhouse-Geisser	Huynh-Feldt	Lower-bound
TIME	MOTIVE	.798	2.938	2	.228	.832	.998	.500
	SKILLS	.954	.618	2	.734	.956	1.000	.500
	READINES	.943	.757	2	.684	.946	1.000	.500
	GRADES	.952	.634	2	.729	.955	1.000	.500
	PERFORM	1.00	.000	0	.	1.000	1.000	1.000

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

b Design: Intercept+SECTION Within Subjects Design: TIME

Box's Test of Equality of Variance-Covariance Matrices[a]

DV	Box's M	F	df1	df2	Sig.
Motivation	14.152	1.801	6	1420.075	.095
Skills	8.854	1.127	6	1420.075	.344
Readiness	7.469	.954	6	1420.075	.455
Grades	8.463	1.077	6	1420.075	.374
Performance	3.070	.802	3	18000.000	.493

Tests the null hypothesis that the observed covariance matrices of the dependent variables are equal across groups

[a] Design: Intercept + METHOD; Within Subjects Design: TIME

Tests of Normality

	SECTION	Shapiro-Wilk		
		Statistic	df	Sig.
MSLQA1	Control Group	.931	8	.497
	Treatment Group	.989	8	.990*
MSLQA2	Control Group	.892	8	.302
	Treatment Group	.913	8	.407
MSLQA3	Control Group	.974	8	.915
	Treatment Group	.944	8	.621
MSLQB1	Control Group	.913	8	.405
	Treatment Group	.898	8	.332
MSLQB2	Control Group	.930	8	.492
	Treatment Group	.964	8	.823
MSLQB3	Control Group	.912	8	.400
	Treatment Group	.929	8	.487
SDLRS1	Control Group	.904	8	.363
	Treatment Group	.948	8	.658
SDLRS2	Control Group	.852	8	.103
	Treatment Group	.922	8	.452
SDLRS3	Control Group	.897	8	.330
	Treatment Group	.980	8	.962
GRADE15	Control Group	.906	8	.374
	Treatment Group	.943	8	.614
PRG6PCT	Control Group	.777	8	.020
	Treatment Group	.985	8	.984
PRG7PCT	Control Group	.704	8	.010**
	Treatment Group	.962	8	.797
SDL2	Control Group	.928	6	.509
	Treatment Group	.903	6	.404
SDL3	Control Group	.806	6	.073
	Treatment Group	.901	6	.398

* This is a lower bound of the true significance.

** This is an upper bound of the true significance.

REFERENCES

Albanese, M. A., & Mitchell, S. (1993). Problem-based learning: A review of literature on its outcomes and implementation issues. Academic Medicine, 68(1), 52-81.

Albion, P. R., & Gibson, I. W. (1998, June 20-25, 1998). Interactive multimedia and problem-based learning: Challenges for instructional design. Paper presented at the ED-MEDIA/ED-TELECOM 98 World Conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecommunications, Freiburg, Germany.

Allen, D. E., Duch, B. J., & Groh, S. E. (1996). The power of problem-based learning in teaching introductory science courses. In W. H. Gijselaers (Ed.), Bringing problem based learning to higher education: theory and practice (Vol. 68, pp. 43-52). San Francisco, CA: Jossey-Bass, Inc.

Arambula-Greenfield, T. (1996). Implementing problem-based learning in a college science class. Journal of College Science Teaching, 26(1), 26-30.

Barrows, H. S. (1986). A taxonomy of problem-based learning methods. Medical Education, 20, 481-486.

Barrows, H. S. (1988). The tutorial process (Revised ed.). Springfield, IL: Southern Illinois University School of Medicine.

Barrows, H. S. (1994). Practice-based learning: Problem-based learning applied to medical education. Springfield, IL: Southern Illinois University School of Medicine.

Barrows, H. S., & Tamblyn, R. M. (1980). Problem-based learning: An approach to medical education. New York, NY: Springer Verlag.

Bitterman, J. A. (1988). Relationship of adults' cognitive style and achieving style to preference for self-directed learning. Unpublished Ed. D., Northern Illinois Unversity.

Blackwood, C. C. (1994). Applying self-directed learning principles in the technical training of a high-risk industry. In R. G. Brockett (Ed.), Overcoming resistance to self-direction in adult learning (Vol. 64, pp. 47-54). San Francisco, CA: Jossey-Bass, Inc.

Bligh, D. A. (2000). What's the use of lectures? San Francisco, CA: Jossey-Bass, Inc.

Blumberg, P., & Michael, J. A. (1992). Development of self-directed learning behaviors in a partially teacher-directed problem-based learning curriculum. Teaching & Learning in Medicine, 4(1), 3-8.

Blumberg, P., Michael, J. A., & Zeitz, H. (1990). Roles of student generated learning issues in problem-based learning. Teaching & Learning in Medicine, 2(3), 149-154.

Blumberg, P., & others, A. (1990, April 16-20, 1990). The uses of student generated learning issues by 7 problem based medical curricula. Paper presented at the Annual Meeting of the American Educational Research Association, Boston, MA.

Bonham, L. A. (1991). Guglielmino's self-directed learning readiness scale: What does it measure? Adult Education Quarterly, 41(2), 91-99.

Borg, W. R., & Gall, M. D. (1989). Educational research: An introduction (5th ed.). New York, NY: Longman.

Brockett, R. G. (1985). The relationship between self-directed learning readiness and life satisfaction among older adults. Adult Education Quarterly, 35(4), 210-219.

Brockett, R. G., & Hiemstra, R. (1991). Self-direction in adult learning: Perspectives on theory, research, and practice. New York, NY: Routledge, Chapman and Hall, Inc.

Brockett, R. G., Stockdale, S. L., Fogerson, D. L., Cox, B. F., Canipe, J. B., Chuprina, L. A., Donaghy, R. C., & Chadwell, N. E. (2000). Two decades of literature on self-directed learning: A content analysis. Paper presented at the International Self-Directed Learning Symposium, Boynton Beach, FL.

Bulik, R. J., & Romero, C. M. (2001). The elusive concept of self-directed learning. In H. B. Long & Associates (Ed.), Self-directed learning and the information age (Vol. 2000 SDL e-Publication). Norman, OK: Oklahoma Research Center for Continuing Professional and Higher Education of the University of Oklahoma.

Caffarella, R. S. (1983). Fostering self-directed learning in post-secondary education: The use of learning contracts. Lifelong Learning, 7(3), 7-10,25-26.

Caffarella, R. S., & Caffarella, E. P. (1986). Self-directedness and learning contracts in adult education. Adult Education Quarterly, 36(4), 226-234.

Campbell, D. T., & Stanley, D. T. (1963). Experimental and quasi-experimentation designs for research. Boston, MA.: Houghton Mifflin Company.

Candy, P. C. (1991). Self-direction for lifelong learning: A comprehensive guide to theory and practice. San Francisco, CA: Jossey-Bass, Inc.

Caplow, J. A. H., Donaldson, J. F., Kardash, C., & Hosokawa, M. (1997). Learning in a problem-based medical curriculum: Students' conceptions. Medical Education, 31(6), 440-447.

Cawley, P. (1997). A problem-based module in mechanical engineering. In G. Feletti (Ed.), The challenge of problem based learning (Revised ed., pp. 185-193). New York, NY: St. Martin's Press.

Charlin, B., Mann, K., & Hansen, P. (1998). The many faces of problem-based learning: A framework for understanding and comparison. Medical Teacher, 20(4), 323-330.

Cheren, M. (1983). Helping learners achieve greater self-direction. In R. M. Smith (Ed.), Helping adults learn how to learn (Vol. New directions for continuing education, pp. 23-38). San Francisco, CA: Jossey-Bass, Inc.

Cockrell, K. S., Hughes Caplow, J. A., & Donaldson, J. F. (2000). A context for learning: Collaborative groups in the problem-based learning environment. The Review of Higher Education, 23(3), 347-363.

Cohen, S. A., & Reese, H. W. (Eds.). (1994). Life-span developmental psychology. Hillsdale, NJ: Lawrence Erlbaum Associates.

Computing Science Accreditation Board. (2001, February 11, 2001). CSAB Home Page [Web Page]. Retrieved March 24, 2001, 2001, from the World Wide Web:

Confessore, G. J. (1991). Human behavior as a construct for assessing Guglielmino's Self-Directed Learning Readiness Scale. In H. B. Long & Associates (Ed.), Self-directed learning: Consensus and conflict (pp. 245-272). Norman, OK: Oklahoma Research Center for Continuing Professional and Higher Education of the University of Oklahoma.

Confessore, G. J., & Confessore, S. J. (1992a). Selecting the key literature in self-directed learning. In S. J. Confessore (Ed.), Guideposts to self-directed learning. Expert commentary on essential concepts (pp. 12-22). King of Prussia, PA: Organization Design and Development, Inc.

Confessore, G. J., & Confessore, S. J. (Eds.). (1992b). Guideposts to self-directed learning. Expert commentary on essential concepts. King of Prussia, PA: Organization Design and Develcpment, Inc.

Cranton, P. (1992). Working with adult learners. Toronto, Canada: Wall & Enerson, Inc.

Dathe, D., O'Brien, K., Loacker, G., & Matlock, M. G. (1997). Learning from the assessment of problem solving. In G. Feletti (Ed.), The challenge of problem-based learning (Revised ed., pp. 294-308). New York, NY: St. Martin's Press.

Dolmans, D., & others, A. (1993, April 12-16, 1993). Validation of a rating scale for tutor evaluation in a problem-based medical curriculum. Paper presented at the Annual Meeting of the American Educational Research Association, Atlanta, GA.

Dolmans, D. H. J. M. (1993). Problem effectiveness in a course using problem-based learning. Academic Medicine, 68(3), 207-213.

Dolmans, D. H. J. M., & Others, A. (1994). Improving the effectiveness of tutors in problem-based learning. Medical Teacher, 16(4), 369-377.

Dolmans, D. H. J. M., & Others, A. (1996). Effects of tutor expertise on student performance in relation to prior knowledge and level of curricular structure. Academic Medicine, 71(9), 1008-1011.

Dolmans, D. H. J. M., Schmidt, H. G., & Gijselaers, H. G. (1995). The relationship between student-generated learning issues and self-study in problem-based learning. Instructional Science, 22, 251-267.

Dolmans, D. H. J. M., Snellen-Balendong, H., Wolfhagen, I. H. A. P., & van der Vleuten, C. P. M. (1997). Seven principles of effective case design for a problem-based curriculum. Medical Teacher, 19(3), 185-189.

Duek, J. E., & Wilkerson, L. (1991). Learning issues identified by students in tutorless problem-based tutorials. Los Angles: Center for Educational Development and Research.

Dunlap, J. C. (1996). The relationship of problem-based learning to life-long learning. Unpublished Ph.D., University of Colorado at Denver, Denver.

Finestone, P. M. (1984). A construct validation of the self-directed learning readiness scale with labour education participants. Unpublished Ph. D., University of Toronto (Canada), Toronto, Canada.

Gijselaers, W. H. (1994, April 4-8, 1994). Analyses of tutor behavior at different time points and within different departments. Paper presented at the Annual Meeting of the American Educational Research Association, New Orleans, LA.

Girden, E. R. (1992). ANOVA repeated measures (Vol. 07-084). Newbury Park, CA: Sage Publications, Inc.

Groh, S. E. (2000, Fall, 2000). The brominator [Web Page]. University of Delaware. Retrieved March, 21, 2001, 2001, from the World Wide Web: http://www.udel.edu/pbl/problems/brominator.html

Grow, G. (1991a). Higher-order skills for professional practice and self-direction. Journalism Educator, 45(4), 56-65.

Grow, G. (1991b). The staged self-directed learning model. In H. B. Long & Associates (Ed.), Self-directed learning: Consensus & Conflict (pp. 199-227): Oklahoma Research Center for Continuing Professional and Higher Education of the University of Oklahoma.

Grow, G. (1991c). Teaching learners to be self-directed. Adult Education Quarterly, 41(3), 125-149.

Grundy, J. C. (1996, January 24-27, 1996). Experiences with facilitating student learning in a group information systems project course. Paper presented at the 1996 International Conference on Software Engineering: Education & Practice, Dunedin, New Zealand.

Guglielmino, L. M. (1977). Development of the self-directed learning readiness scale. Unpublished Doctor of Education, University of Georgia, Athens, Georgia.

Guglielmino, L. M. (1989). Reactions to Field's investigation into the SDLRS. Adult Education Quarterly, 39(4), 235-245.

Guglielmino, L. M. (1997). Reliability and validity of the self-directed learning readiness scale and the learning preference assessment. In H. B. Long & Associates (Ed.), Expanding horizons in self-directed learning (pp. 209-220). Norman, OK: Oklahoma Research Center for Continuing Professional and Higher Education of the University of Oklahoma.

Guglielmino, L. M., & Guglielmino, P. J. (1994). Practical experience with self-directed learning in business and industry human resource development. In R. G. Brockett (Ed.), Overcoming resistance to self-direction in adult learning (Vol. 64, pp. 39-46). San Francisco, CA: Jossey-Bass, Inc.

Hair, J. F., Jr., Anderson, R. E., Tatham, R. L., & Black, W. C. (1998). Multivariate data analysis (5th ed.). Upper Saddle River, NJ: Prentice-Hall, Inc.

Hall-Johnsen, K. J. (1985). The relationship between readiness for, and involvement in, self-directed learning. Unpublished Ph. D., Iowa State University.

Hartman, J., & White, C. M. (1990). "Real world" skills vs. "school taught" skills for the undergraduate computer major. Paper presented at the 21st SIGCSE Technical Symposium on Computer Science Education.

Hassan, A. M. (1981). An investigation of the learning projects among adults of high and low readiness for self-direction in learning. Unpublished Ph. D., Iowa State University.

Hertzog, C. (1994). Repeated measures analysis in developmental research: What our ANOVA text didn't tell us. In H. W. Reese (Ed.), Life-span developmental psychology (pp. 187-222). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hertzog, C., & Rovine, M. (1985). Repeated measures analysis in developmental research: Selected issues. Child Development, 56(4), 787-809.

Hiemstra, R. (1992). Individualizing the instructional process: What we have learned from two decades of research on self-direction in learning. In H. B. Long & Associates (Ed.), Self-directed learning: Application and Research (pp. 323-344). Norman, OK: Oklahoma Research Center for Continuing Professional and Higher Education of the University of Oklahoma.

Hmelo, C. E., & Ferrari, M. (1997). The problem-based learning tutorial: Cultivating higher order thinking skills. Journal for the Education of the Gifted, 20(4), 401-422.

Hmelo, C. E., Gotterer, G. S., & Bransford, J. D. (1997). A theory-driven approach to assessing the cognitive effects of PBL. Instructional Science, 25(6), 387-408.

Hoban, G., Sersland, C., & Raine, B. (2001). Can adult learners raise their self-efficacy for self-directed learning? A reflective challenge to some of our assumptions. In H. B. Long & Associates (Ed.), Self-directed learning and the information age (Vol. 2000 SDL e-Publication). Norman, OK: Oklahoma Research Center for Continuing Professional and Higher Education of the University of Oklahoma.

Houle, C. O. (1961). The inquiring mind: A study of the adult who continues to learn. Madison, WI: University of Wisconsin Press.

Houle, C. O. (1988). The inquiring mind: A study of the adult who continues to learn (2nd ed.). Norman, OK: Oklahoma Research Center for Continuing Professional and Higher Education, University of Oklahoma.

Hrimech, M. (1995). Some self-regulated learning strategies utilized by advanced adult learners. In H. B. Long & Associates (Ed.), New dimensions in self-directed learning (pp. 87-97). Norman, OK: Oklahoma Research Center for Continuing Professional and Higher Education of the University of Oklahoma.

Humphrey, W. S. (1997). Introduction to the personal software process. Reading, MA: Addison-Wesley Publishing Co.

Johnson, D. W., & Johnson, R. T. (1999). Learning together and alone (5th ed.). Boston, MA.: Allyn and Bacon.

Jonassen, D. H., & Kwon, H. (2001). Communication patterns in computer mediated versus face-to-face group problem solving. Educational Technology Research and Development, 49(1), 35-51.

Jones, C. J. (1989). A study of the relationship of self-directet learning readiness to observable behavioral characteristics in an adult basic education program. Unpublished Ed. D., University of Georgia.

Knowles, M. S. (1975). Self-directed learning: A guide for learners and teachers. Englewood Cliffs, NJ: Prentice Hall Regents.

Knowles, M. S. (1990). The adult learner: A neglected species (4th ed.). Houston, TX: Gulf Publishing Company.

Koschmann, T., Kelson, A. C., Feltovich, P. J., & Barrows, H. S. (1996). Computer-supported problem-based learning: A principled approach to the use of computers in collaborative learning. In T. Koschmann (Ed.), CSCL: Theory and practice of an emerging paradigm (pp. 83-124). Mahwah, NJ: Lawrence Erlbaum Associates.

Krzanowski, W. J. (2000). Principles of multivariate analysis: A user's perspective (Revised ed.). New York, NY: Oxford University Press.

Lieux, E. M. (1996). The effect of teaching method on student's knowledge of quantity food production and service. course evaluations, and propensity for participative management. Unpublished Ph.D., Virginia Polytechnic Institute and State University.

Loats, J. T. (2001). Students' adjustments to problem-based learning.

Long, H. B. (1987). Item analysis of Guglielmino's self-directed learning readiness scale. International Journal of Lifelong Education, 6(4), p331-336.

Long, H. B. (1990). Changing concepts of self-direction in learning. In H. B. Long & Associates (Ed.), Self-directed learning: Emerging theory and practice. Norman, OK: Oklahoma Research Center for Continuing Professional and Higher Education of the University of Oklahoma.

Long, H. B. (2001). A multi-variable theory of self-direction in learning. In H. B. Long & Associates (Ed.), Self-directed learning and the information age (Vol. 2000 SDL e-Publication). Norman, OK: Oklahoma Research Center for Continuing Professional and Higher Education of the University of Oklahoma.

Long, H. B., & Agyekum, S. K. (1983). Guglielmino's self-directed learning readiness scale: A validation study. Higher Education, 12(1), 77-87.

Long, H. B., & Agyekum, S. K. (1984). Teacher ratings in the validation of Guglielmino's self-directed learning readiness scale. Higher Education, 13(6), 709-715.

McCune, S. K. (1988). A meta-analytic study of adult self-direction in learning: A review of the research from 1977 to 1987. Unpublished Ph. D., Texas A&M University, College Station, TX.

Mehrens, W. A., & Lehmann, I. J. (1984). Measurement and evaluation in education and psychology (3rd ed.). New York, NY: Holt, Rinehart, and Winston.

Metropolitan State College of Denver. (2001, 3/23/01). College catalog: Course descriptions [Web Page]. Retrieved March 24, 2001, 2001, from the World Wide Web: http://www.mscd.edu/academic/catalog/courses/coursesA-H.htm

Moust, J. H. C., & Schmidt, H. G. (1995). Facilitating small-group learning: A comparison of student and staff tutors' behavior. Instructional Science, 22, 287-301.

National Science Foundation Advisory Committee. (1998). Shaping the future: New expectations for undergraduate education in science, mathematics, engineering, and technology (NS 1.2: F 98/2/ V.1). Washington, D.C.: National Science Foundation.

Norman, G. R. (1997). Assessment in problem-based learning. In G. Feletti (Ed.), The challenge of problem based learning (Revised ed., pp. 263-268). New York, NY: St. Martin's Press.

O'Brien, R. G., & Kaiser, M. K. (1985). MANOVA method for analyzing repeated measures designs: An extensive primer. Psychological Bulletin, 97(2), 316-333.

Pilling-Cormick, J. (1994). Resistance by educators to using self-directed learning perception scale. In R. G. Brockett (Ed.), Overcoming resistance to self-direction in adult learning (Vol. 64, pp. 63-70). San Francisco, CA: Jossey-Bass, Inc.

Pilling-Cormick, J. (1996). Development of the self-directed learning perception scale. Unpublished Ph.D., University of Toronto (Canada), Toronto, Canada.

Pintrich, P. R. (1995). Understanding self-regulated learning, Understanding self-regulated learning (Vol. 63, pp. 3-12). San Francisco, CA: Jossey-Bass, Inc.

Pintrich, P. R., & DeGroot, E. (1990). Motivation and self-regulated learning components of classroom academic performance. Journal of Educational Psychology, 82, 33-40.

Pintrich, P. R., & Schunk, D. H. (1996). Motivation in education: theory, research, and applications. Englewood Cliffs: Prentice-Hall, Inc.

Pintrich, P. R., Smith, D. A. F., Garcia, T., & McKeachie, W. J. (1991). A manual for the use of the motivated strategies for learning questionnaire (MSLQ) (Technical Report No. 91-B-004). Ann Arbor, MI: University of Michigan.

Pintrich, P. R., Smith, D. A. F., Garcia, T., & Mckeachie, W. J. (1993). Reliability and predictive validity of the motivated strategies for learning questionnaire (MSLQ). Educational and psychological measurement, 53(3), 801-814.

Pomberg, G. (1993). Software engineering education--adjusting our sails. Education and Computing, 8(4), 287-294.

Ponton, M. K., Carr, P. B., & Confessore, G. J. (2000). Learning conation: A psychological perspective of personal initiative and resourcefulness. In H. B. Long & Associates (Ed.), Practice & theory in self-directed learning (pp. 65-82). Schaumburg, IL: Motorola University Press.

Ram, P. (1999). Problem-based learning in undergraduate education. Journal of Chemical Education, 76(8), 1122-1126.

Rangachari, P. K. (1996). Twenty-Up: Problem-based learning with a large group. In W. H. Gijselaers (Ed.), Bringing problem based learning to higher education: theory and practice (Vol. 68, pp. 63-71). San Francisco, CA: Jossey-Bass, Inc.

Rutland, A. M., & Guglielmino, L. M. (1987). Increasing readiness for self-directed learning: A facilitator's manual for ten self-directed learning group modules for adults. Boca Raton, Florida: Florida Atlantic University.

Ryan, G. (1993). Student perceptions about self-directed learning in a professional course implementing problem-based learning. Studies in Higher Education, 18(1), 53-63.

Schmidt, H., van der Arend, A., Kokx, I., & Boon, L. (1995). Peer versus staff tutoring in problem-based learning. Instructional Science, 22, 279-285.

Schmidt, H. G. (1983). Problem-based learning: rational and description. Medical Education, 17, 11-16.

Schmidt, H. G., & Moust, J. H. C. (1995, April 18-22, 1995). What makes a tutor effective? A structural equations modeling approach to learning in problem-based curricula. Paper presented at the Annual Meeting of the American Educational Research Association, San Francisco, CA.

Schmidt, H. G., & Others, A. (1993). Influence of tutors' subject-matter expertise on student effort and achievement in problem-based learning. Academic Medicine, 68(10), 784-791.

Segers, M. S. R. (1997). An alternative for assessing problem-solving skills: The overall test. Studies in Educational Evaluation, 23(4), 373-398.

Shaw, M. (1976). Making software engineering issues real to undergraduates. In P. Freeman (Ed.), Software engineering education (pp. 104-107). New York, NY: Springer Verlag.

Shaw, M. (2000, June 4 - 11, 2000). Software engineering education: A roadmap. Paper presented at the International Conference on Software Engineering, Limerick, Ireland.

Skaggs, B. J. (1981). The relationship between involvement of professional nurses in self-directed learning activities, loci of control, and readiness for self-directed learning measures. Unpublished Ph. D., University of Georgia, Austin, TX.

SPSS. (1999). Statistical Package for the Social Sciences (SPSS) (Version 10.05). Chicago, Illinois: SPSS, Inc.

Stepien, W. J., & Pyke, S. L. (1997). Designing problem-based learning units. Journal for the Education of the Gifted, 20(4), 380-400.

Stepien, W. J., Senn, P. R., & Stepien, W. C. (2000). The Internet and problem-based learning: Developing solutions through the Web. Tucson, AZ: Zephyr Press.

Stipek, D. (1998). Motivation to learn: From theory to practice (3rd ed.). Boston, MA.: Allyn and Bacon.

Swanson, D. B., Case, S. M., & van der Vleuten, C. P. (1997). Strategies for student assessment. In G. Feletti (Ed.), The challenge of problem based learning (Revised ed., pp. 269-282). New York, NY: St. Martin's Press.

Tabachnick, B. G., & Fidell, L. S. (2001). Using multivariate Statistics (Fourth ed.). New York, NY: HarperCollins College Publishers.

Taylor, M. (1986). Learning for self-direction in the classroom: The pattern of a transition process. Studies in Higher Education, 11(1), 55-72.

Todd, S. (1997). Preparing tertiary teachers for problem-based learning. In G. Feletti (Ed.), The challenge of problem based learning (Revised ed., pp. 117-124). New York, NY: St. Martin's Press.

Tough, A. (1978). Major learning efforts: Recent research and future directions. Adult Education, 28(4), 250-263.

Vernon, D. T. A., & Blake, R. L. (1993). Does problem-based learning work? A meta-analysis of evaluative research. Academic Medicine, 68(7), 550-563.

Vygotsky, L. S. (1978). Mind in society: The development of higher psychological processes. Cambridge, MA.: Harvard University Press.

Wegner, S. B., Holloway, K., & Crader, A. (1997). Utilizing a problem-based approach on the World Wide Web.

Wilkerson, L. (1995). Identification of skills for the problem-based tutor: Student and faculty perspectives. Instructional Science, 22, 303-315.

Wilkerson, L. (1996). Tutors and small groups in problem-based learning: Lessons from the literature. In W. H. Gijselaers (Ed.), Bringing problem based learning to higher education: theory and practice (Vol. 68, pp. 23-32). San Francisco, CA: Jossey-Bass, Inc.

Wilkerson, L., & Hundert, E. M. (1997). Becoming a problem-based tutor: Increasing self-awareness through faculty development. In G. Feletti (Ed.), The challenge of problem-based learning (Revised ed., pp. 160-172). New York, NY: St. Martin's Press.

Wilson, J. D., Hoskin, N., & Nosek, J. T. (1993, February 18 - 19, 1993). The benefits of collaboration for student programmers. Paper presented at the 24th SIGCSE Technical Symposium on Computer Science Education, Indianapolis, IN.

Woods, D. R. (1996). Problem-based learning for large classes in chemical engineering. In W. H. Gijselaers (Ed.), Bringing problem based learning to higher education: theory and practice (Vol. 68, pp. 91-99). San Francisco, CA: Jossey-Bass, Inc.