5-2021

The Use of Team-Based Learning to Improve Unit Test Scores of Adult Learners in Introduction to Programming Courses in a Small Southwest Georgia University

Karen Streetman Cook

The Use of Team-Based Learning to Improve Unit Test Scores of Adult Learners in Introduction to Programming Courses in a Small Southwest Georgia University

by Karen Streetman Cook

This dissertation has been read and approved as fulfilling the partial requirement for the Degree of Doctor of Education in Curriculum and Leadership.

____
Jennifer L. Brown, PhD
Chair and Methodologist

____
Jennifer M. Lovelace, PhD
Director, Doctoral Program in Education

____
Yesem Peker, PhD
Committee Member

____
Brian Tyo, PhD
Director, COEHP Graduate Studies

____
Kimberly Shaw, PhD
Committee Member

____
Deirdre Greer, PhD
Dean, COEHP

THE USE OF TEAM-BASED LEARNING TO IMPROVE UNIT TEST SCORES OF ADULT LEARNERS IN INTRODUCTION TO PROGRAMMING COURSES IN A SMALL SOUTHWEST GEORGIA UNIVERSITY

by

Karen Streetman Cook

A Dissertation
Submitted in Partial Fulfillment of
the Requirements for
the Degree of Doctor of Education
in Curriculum and Leadership
(CURRICULUM)

Columbus State University
Columbus, GA

May 2021

DEDICATION

I would like to dedicate my dissertation to God for giving me the courage and strength to endure and for providing the support and encouragement of the following people to help me through this journey.

To my parents, Burnette and Carol Streetman, for all of their support and their confidence in me that I could accomplish my dream of earning my doctoral degree. They are the strongest and most dedicated people that I have ever known, and I am so thankful that they are mine.

To my best friend, Monique Daniels, for never letting me give up and having faith in me even when I did not have faith in myself. I could not have made it this far without her love and support.

To my friend and mentor, Dr. Frank Jones. Dr. Jones was not only my advisor when I first began my college career, but he has been my mentor throughout all these years. He is the reason that I wanted to earn my doctoral degree from the time that I first started college and has been the encouragement that I have needed to finish. I am thankful to have had such a wonderful role model in my life.

I would also like to thank my children, Addie, Ben, Sarah, and Zach, for their patience with me while going to school and working full-time. I hope that you always have the desire to follow your dreams.

I love you all with all of my heart.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my dissertation chair, Dr. Jennifer Brown, for all of her guidance, knowledge, patience, and motivation to help me complete my dissertation. I would not have been able to complete my dissertation without all of her help and support. I would also like to thank my committee members, Dr. Yesem Peker and Dr. Kimberly Shaw, for their guidance and support.

ABSTRACT

Low success rates in the introduction to programming course at a small southwest Georgia state university have led to a low retention rate in the computer science degree program. The instructional strategy used to teach the course could contribute to the low success rates of the students in the course. Andragogy, the adult learning theory, was used as the theoretical framework for this study. The purpose of this quantitative causal-comparative study was to determine if differences existed in the numerical unit test scores between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format. The participants included 52 students who took the introduction to programming course during Spring 2019 and Fall 2019. The students' unit test scores were analyzed using a series of ANCOVAs. No statistically significant differences were found in the unit test scores between the students who took introduction to programming using a traditional lecture and lab-based format and the students who took introduction to programming using a team-based learning format. By introducing the students to team-based learning earlier in the introduction to programming course, students' confidence in their ability to make decisions could increase, course withdrawal rates could decrease, and the students' ability to work in teams could be improved.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Background of the Problem

The need for computer programmers is expected to grow 12% from 2018 to 2028 (Bureau of Labor Statistics, 2019). In 2012, research conducted by Microsoft indicated that U.S. colleges were not producing enough computer science graduates to meet the expected demand by 2020 (Microsoft, 2012). According to the Smithsonian Science Education Center (n.d.), a projected 2.4 million science, technology, engineering, and math (STEM) jobs would not be filled by 2018. Students typically will begin their study in computer science degree programs by taking an introduction to programming course. Programming courses are used to teach problem-solving skills using a programming language chosen by each individual university. Programming knowledge and logical thinking are critical skills required to be successful in computer science. The introductory courses help students from different backgrounds in computing to start on an even level (Joint Task Force on Computing Curricula, 2013).

According to Bennedsen and Casperson (2007), an average failure rate of 33% was reported for introduction to programming courses. Bennedsen and Casperson (2019) replicated their study and found a slight improvement with a reported 28% failure rate. Low success rates in the introduction to programming course could be one of the reasons for the lack of graduates.

The instructional strategy that has traditionally been used to teach introduction to programming courses has been a lecture and lab-based format. The instructional strategy used to teach the course could be one of the reasons the course has low success rates

(Canedo, Almeida, & Leite, 2018). Previous research has shown that the students' failure can be related to the instructional strategy used to teach the class (Alturki, 2016). Finding an instructional strategy that helps the students learn programming better could improve the success rate of the students in the introduction to programming course (Vihavainen, Airaksinen, & Watson, 2014).

Team-based learning has been implemented successfully in over 200 campuses in the United States and in eight different countries in various courses (Team-Based Learning Collaborative, n.d.). Comeford (2016) reported an improvement in the success rate for introduction to chemistry courses taught using team-based learning. An attrition rate of 31% was reported between Fall 2008 to Fall 2009 when the course that was taught using a traditional lecture and lab-based class format. When the course was taught using a team-based learning format between Spring 2010 and Spring 2013, the attrition rate fell to 19% (Comeford, 2016). More research is needed in other science courses, such as introduction to computer programming courses, to determine if using the team-based learning instructional strategy could increase academic performance, which could improve the success rate for these courses.

Statement of the Problem

A problem exists in the STEM degree programs, particularly in computer science and information technology, because there is a low success rate in the introduction to programming course, which is the gateway course for four STEM degree programs (i.e., computer science, information technology, mathematics, and engineering). That problem, specifically, is the low success rate in the introduction to programming course at a small southwest Georgia university. Currently, much research has been conducted on the causes of these low success rates (e.g., Alturki, 2016; Bosse & Gerosa, 2017; Hawi, 2010; Hegazi & Alhawarat, 2015); however, little research has been conducted on how to improve these success rates specifically. This problem impacts the growing demand of computer science graduates because the low success rates in the introduction to programming courses lead to a low retention rate in the computer science degree majors (Chen, 2015). Many possible factors contribute to this problem, including poor advising, a lack of math skills, poorly designed introduction to programming courses, insufficient practice and feedback, poor graduate student instructors, poor management abilities, and the choice of language in addition to when object-oriented programming was introduced (Beaubouef & Mason, 2005). Of the factors found in Beaubouef and Mason's (2005) study, team-based learning could improve the problem of having poorly designed courses. Courses should be designed and planned carefully to implement team-based learning properly (Comeford, 2016). In addition, because team-based learning requires the students within the course to design, write, and test computer programs together, the students would have more practice writing code and learning to manage their time and shared skills (Wu, Farquhar, & Compton, 2018). The individual and team readiness assurance tests also provide immediate feedback on the concepts that the students are learning (Burgess et al., 2017). This study will contribute to the body of knowledge needed to address this problem by examining the use of team-based learning in an introduction to programming course, which could increase the success rates of computer science majors.

Purpose of the Study

The purpose of this study was to determine if differences existed in the numerical unit test scores between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format. An improvement in the unit test scores could improve success rates of the students in the introduction to programming course and increase the retention rates of students in the computer science major. The independent variable was defined as the group (i.e., the control group was taught using a traditional lecture and lab class and the treatment group was taught using team-based learning). The dependent variable was the numerical unit test scores in the course. The unit tests covered the topics of conditional expressions and looping expressions. The covariates were the unit tests that covered general programming terminology and data types and methods and behaviors.

Research Questions and Hypotheses

The research questions for this study were:

RQ1. What is the difference in students' knowledge of conditional expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university?

H_{01}: There is no difference in students' knowledge of conditional expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university to a statistically significant degree.

H_{a1}: There is a difference in students' knowledge of conditional expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university to a statistically significant degree.

RQ2. What is the difference in students' knowledge of looping expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university?

H_{02}: There is no difference in students' knowledge of looping expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university to a statistically significant degree.

H_{a2}: There is a difference in students' knowledge of looping expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university to a statistically significant degree.

Theoretical Framework

Andragogy

Malcom Knowles is credited with developing the adult learning theory referred to as andragogy (Pappas, 2013). Andragogy refers to the different ways that adults learn compared to the ways that children learn. Adults are more inclined to be self-directed, motivated, and willing to learn. Andragogy, or the adult learning theory, was introduced in 1968 by Knowles. A large volume of research had been conducted on ways to teach children, known as pedagogy, but very little, if any, research had been conducted on how to teach adults (Chen, 2015).

Knowles defined pedagogy as a teacher-facilitated teaching style that focuses on content rather than on solving problems. Pedagogy is the teaching style that is normally used for teaching children. In pedagogy, the teacher controls every part of the students' learning. In the case where the students have no knowledge of the subject and are young, teachers tend to prefer pedagogy (Knowles, 1984). Several key differences exist between pedagogy and andragogy. One of the differences is that children are dependent learners and adults are self-directed learners. Children do not have experiences that they have learned from, whereas adults have these experiences. In addition, children are required to attend school, and adults go to school on their own when they want to learn a specific skill or knowledge. The fourth difference is that children learn so that they can use the knowledge that they gain when they are older and adults learn to improve their current situation now (Muduli, Kaura, & Quazi, 2018).

With adult learning, the students are active participants in the classroom, and teachers are passive participants. The teachers serve as facilitators in the process of learning rather than just being the source of information. Applying andragogy may provide a more effective method of teaching and learning for students in the introduction to programming course. Andragogy may also produce a more long-term acquirement of knowledge and skills (Malik & Khaliq, 2017).

Typically, adult learners are defined as students who are 25 years old and older. These students are also known as non-traditional students. However, adult learners also include students who are under 25 years old and have adult responsibilities, such as working full-time, living on their own, and having children, as well as those students who did not go directly into higher education after high school or those students who did not complete high school. According to this definition, the majority of students in higher education could be considered adult learners. Traditional students are those students who live on campus, are enrolled as full time students, and do not have outside responsibilities (Chen, 2017). Having to manage the multiple roles that adult learners have makes it difficult for them to find time to study and prepare for classes or to participate in campus-based organizations and activities (Ross-Gordon, 2011).

The target population for this study included students who were enrolled in on-campus classes or a combination of on-campus and online classes. The population consisted of 60.4% of students who were commuter students, 47.2% of students who were first-generation students, and 40.6% of students who received the Pell grant. The average age of all of the undergraduate students at the university where the research was conducted was 23.7 years of age. Part-time undergraduate students accounted for 34.45% of the total number of undergraduate students (College, n.d.b). In addition, most of the students, both residential students and commuter students, were considered non-traditional students because they had jobs, children, and responsibilities outside of their academic studies. The National Center for Educational Statistics found that the enrollment of students aged 25 and over increased 11% from 2006 to 2016 and is expected to continue to increase (U.S. Department of Educational, 2019).

The theory of pedagogy includes students who are in pre-kindergarten through their 12th grade of high school. These students typically include ages 18 and under (Knowles, 1984). The theory of andragogy includes students that are age 25 and older (Pappas, 2013). A learning theory that is devoted to college students between these two age groups does not exist currently. Because the students in this research study fit more closely with adult learners who make the choice themselves to attend school and select for themselves what they want to study, the theory of andragogy was used for the research. The continuum of how students are taught beginning in pre-kindergarten throughout high school and into university level courses should progress from a pedagogical approach to an andragogical approach given the learner-centered approach of andragogy produces deeper learning. A goal of higher education should be to teach students to be independent learners, which is one of the goals of andragogy and team-based learning (Knowles, 1973).

Team-based Learning

Larry Michaelsen is credited with the development of team-based learning in the 1970s while he was working at The University of Oklahoma. Michaelsen's small group-based instructional process was developed specifically for use with teaching adults in a college classroom setting. This process is being used successfully in over 200 campuses in the United States and in eight different countries (Michaelsen, Knight, & Fink, 2004; Team-Based Learning Collaborative, n.d.). Team-based learning is a four-step process that promotes student learning and participation before and during class by increasing the students' engagement and the amount of knowledge that they retain (Najdanovic-Visak, 2017). The four stages are 1) student preparation, 2) readiness assurance, 3) application, and 4) peer assessment. Lasserre and Szostak (2011) found that using team-based learning to teach these courses increased the completion and pass rate of introduction to programming courses by 20% as compared to the same course being taught with a traditional lecture-based format.

Methodology Overview

The research design for the study was a causal-comparative research design. A causal-comparative design examines the differences between the independent and the dependent variables after an event has happened. The purpose of a causal-comparative research design is to determine if the independent variable had an effect on the dependent variable by comparing two or more groups of individuals (Salkind, 2010). The independent variable was the group (i.e., control group and treatment group). The control group included students who were taught using a traditional lecture-based teaching strategy, and the treatment group included the students who were taught using the team-based learning teaching strategy. The dependent variables were the numerical unit test scores over conditional expressions and the numerical unit test scores over looping expressions.

For the study, the researcher used pre-existing unit test scores from the introduction to programming course section that was taught in the Fall 2018 and the Spring 2019 using a traditional lecture and lab-based teaching strategy for the control group. The researcher used the pre-existing unit test scores from the introduction to programming course section that was taught in the Fall 2019 semester using team-based learning for the treatment group. The population for the study included the 128 computer science majors for Fall 2018, the 108 computer science majors for Spring 2019, and the 159 computer science majors for Fall 2019. The sample included all of the students who took the introduction to programming course during the Fall 2018, Spring 2019, or Fall 2019 semesters and who completed all four of the unit tests. Eighteen students were included for the Fall 2018 section, 28 students were included for the Spring 2019 section, and 24 students were included for the Fall 2019 section. The data for the Fall 2018 section were removed from the analysis due to the unequal variance, which violated the assumption of homogeneity.

For the Fall 2019 section, the four-stage team-based learning approach was used (Jeno et al., 2017). The first stage of team-based learning required students, who had been assigned to a team, to prepare for class by reading the content or by watching a pre-recorded lecture over the content before coming to class. The readiness assurance process included the individual readiness assurance test. The individual readiness assurance test was a multiple-choice test given at the beginning of the class to determine how much the student learned from the pre-class preparation exercise. After each student had taken the individual readiness assurance test independently, the multiple-choice test was taken together by the teams. The team readiness assurance test used an immediate-feedback assessment technique using the quiz tool in the learning management tool provided by the university. The next stage was application where each team completed an assignment by applying the knowledge that they have learned. Each team was assigned the same assignment. The teams then gave their answers simultaneously (Jeno et al., 2017).

The students' numerical unit test scores covering programming terminology and data types and the numerical unit test scores covering methods and behaviors served as the covariate variables to determine if the students' foundational knowledge affected the students' performance on the numerical unit test scores during the second half of the semester. The students' numerical unit test scores covering conditional expressions and looping expressions were compared using an analysis of covariance (ANCOVA) test in SPSS. The independent variable was the group (i.e., control group and treatment group). The control group included the students who were taught using a traditional lecture-based teaching strategy, and the treatment group included the students who were taught using the team-based learning teaching strategy. The dependent variables were the numerical unit test scores over conditional expressions and the numerical unit test scores over looping expressions.

Delimitations and Limitations

This study was conducted in a small university for a required course for the students' major, so the results of the research may not be generalized to larger universities or elective courses. A limitation of the study was the small pre-existing sample. Another limitation was that the semester used for the study was the first time that team-based learning was taught by the instructor and used by the students. Over time, the instructor and the students might become more familiar with the instructional strategy, which could affect the academic performance of the students. Another limitation of the study was the unexpected COVID-19 pandemic that occurred at the midpoint of Spring 2020, when all university system schools in the state were required to complete the semester virtually. Due to the transition from in-class instruction to virtual instruction, the Spring 2020 introduction to programming course could not utilize the same intervention procedures as Fall 2019, and the data from that section could not be included in this study..

Definition of Terms

Adult learners - students who are usually employed, at least 25 years old, in the process of earning a degree or changing careers, and having more life experience than traditional students (Conner, Richardson, & Murphy, 2019). Adult learners also include students who are under 25 years old who have adult responsibilities, such as working full-time, living on their own, and having children, as well as those students who did not go directly into higher education after high school or students who did not complete high school (Chen, 2017). Because no specific term is given to students who have completed high school and who are no longer child learners, but are not by definition adult learners, for the purpose of this study, the term, adult learner, included students who have graduated high school and are pursuing a degree.

Andragogy –the learning theory that outlines the different ways that adults learn in comparison to the ways that children learn (Knowles, 1980).

First-year students - freshman students who are taking university-level courses for the first time and have earned less than 30 semester credit hours (College, n.d.c, p. 58).

Introduction to programming course - the first course that a computer science major takes in their degree program (Sultana & Reid, 2017).

Pedagogy –a teaching style that is used to teach children (Knowles, 1984).

Retention – the ability of the degree program being able to retain the student in the program (Aljohani, 2016).

Second-year students - students who are in their second year of taking courses in a university and have earned less than 60 semester credit hours (College, n.d.c, p. 58).

Success rate - determined by the student earning an A, B, or C in the introduction to programming course (College, n.d.a).

Team-based learning - a teaching strategy designed around modules that include preparation, readiness testing, and application development. Team-based learning helps to teach teamwork and active learning (Pardamean et al., 2017).

Significance of the Study

The results of the study helped to determine if the use of team-based learning could improve the unit test scores of the students in the introduction to programming course. If using team-based learning improves the success rate of the introduction to programming course, the number of students who stay in the affected STEM programs (i.e., computer science, information technology, mathematics, and engineering) could improve, which could lead to an increased graduation rate. More graduates in computer science could help fill the growing number of jobs in this field and could help fill the expected needs (Bureau of Labor Statistics, 2015; Microsoft, 2012). Although the study did not find that team-based learning improved the numerical unit test scores in the introduction to programming course with the small sample size, benefits have been found in studies that included larger sample sizes. Remington et al. (2017) found that team-based learning improved the test scores of the students and gave the students more confidence in their ability to make decisions after taking the course using team-based learning. Similarly, Lasserre and Szostak (2011) found that team-based learning reflected major improvements in the withdrawal rate and the success of the students. With improved training and a larger sample size, similar results may be possible for an introduction to programming course. Improving the success rate in the introduction to programming course could help improve the attrition rate in the computer science degree program and improve the graduation rate in the university.

Summary

Introduction to programming courses have reported to have a low student success rate (Bennedsen & Casperson, 2019). Research has shown that the instructional strategy used to teach the introduction to programming course could have an impact on the student success rate (Canedo et al., 2018). When teaching adult learners using the team-based learning instructional strategy, the success rate of the students taking the course has been improved (Comeford, 2016). The purpose of this study was to determine if differences existed in the numerical unit test scores between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format. An improvement in the unit test scores could improve the success rates of students in the introduction to programming course and could increase the retention rates of students in the computer science major.

CHAPTER II

REVIEW OF LITERATURE

Although a large number of students enter the computer science degree fields, only a small number actually graduate. Large percentages of students either fail or withdraw during the first programming course. Watson and Li (2014) found an average pass rate of only 67.7% in their study of 161 introduction to programming courses in 51 universities from 15 different countries. Bennedsen and Caspersen (2019) found a similar pass rate of only 72% in their study. Although much research has been conducted on the reasons and causes of low retention rates in the computer science degree field, much less research has been conducted on examining teaching strategies that have the potential to improve the problem. A gap in the literature examining teaching strategies to improve student success in the introduction to programming course currently exists. This study aimed to examine an approach to teaching the introduction to programming course that could contribute to that gap in the literature.

This review of literature included the previous research that has been completed in the areas that contribute to the current study. The adult learning theory was used as the theoretical framework of the current study. Introduction to programming has been taught using a traditional lecture and lab-based instructional strategy. This strategy has been found in the literature to lead to low success rates (Bennedsen & Caspersen, 2007, 2019). The literature shows that low success rates of the students in the introduction to programming course lead to low retention rates in computer science (Chen, 2015). Because the students in the introduction to programming course are considered adults and have different learning styles than children, the traditional lecture and lab-based instructional strategy may not be the most appropriate instructional strategy to use. The team-based learning instructional strategy was created specifically for the use in adults (Najdanovic-Visak, 2017). Team-based learning has been found to improve the success rates in the introduction to programming course. In addition, the increased success rates in the introduction to programming course has helped to increase the retention rate in the computer science degree program at The University of British Columbia, Okanagan Campus (Lasserre & Szostak, 2011).

Theoretical Framework

Although adults have been furthering their education since the beginning of time, only after the founding of the American Association for Adult Education in 1926 was adult education officially acknowledged (Knowles, 1980). Since its founding, adult education has continued to grow. The adult educator is more than just a school teacher. Other adult educators include chairpersons of programs, group leaders, program directors, executors, and administrators. The mission of the adult educator is to meet the needs and the goals of individuals, institutions, and society. The adult educator helps the adult student to realize that learning does not end but is instead a lifelong process and that the student needs to learn the skills to be able to be a self-directed learner. A crucial part of adult learning is for the learner to know how to look for answers on his or her own. The adult educator's job is not to give the adult learner the answers but to assist him or her in finding the answers himself or herself. Although the term andragogy originally came from the Greek word *anêr*, which means "man, not boy", and was first used by European educators, Malcolm Knowles is credited with creating the adult learning theory known as andragogy (Knowles, 1980).

Knowles learning theory, andragogy, is based on five assumptions of adult students (Pappas, 2013). The first assumption is self-concept. Unlike a child who has a dependent personality, as a person becomes an adult, his or her self-concept should become more self-directed. The second assumption is the adult student experience. As a person grows older, he or she begins to accumulate experiences that he or she can use to make future decisions. The third assumption is readiness to learn. As a person grows older, his or her readiness to learn naturally leans towards what he or she needs for his or her social role(s) in life. The fourth assumption is orientation to learning. As a person grows older, he or she begins to apply the knowledge that he or she has been learning. He or she changes his or her focus from learning about subjects to learning how to solve problems. The fifth assumption is motivation to learn. As a person grows older, the motivation to learn becomes an internal drive for the person's self-satisfaction in contrast to a child's desire to learn to satisfy someone else. Knowles also suggested four principles of andragogy. The first principle is adults need to take part in planning and evaluating their education. The second principle is the experiences that an adult has, including his or her mistakes, are the foundation for learning activities. The third principle is adults are the most concerned with learning the information that is relevant immediately to what affects their current job or situation. The fourth principle is adult learning is centered on problem solving and not just learning content (Pappas, 2013).

In andragogy, the responsibility of learning is the adult learner's responsibility. The student in this case is motivated internally and more concerned with why the information is important than just memorizing facts (Celli & Young, 2017). In their research, Celli and Young (2017) conducted a qualitative survey of small independent universities to determine ways to improve adult learning in the 21st century. They determined that faculty should take into consideration the students' needs for being successful in their current learning but also prepare them for their future career. The students need to learn how to work in groups and how to learn from their experiences. The faculty should include these experiences in the classroom exercises. For these experiences to be successful, they need to be well-planned and evidence-based. While completing these experiences, the adult learner should find strategies that he or she can use based on his or her own personal strengths and weaknesses. Being able to control his or her behaviors and emotions while completing an exercise is a critical skill for an adult learner. Being able to self-regulate, be persistent, and know when to ask for help add to the success of the adult learner (Celli & Young, 2017).

The National Center for Education Statistics found that 38% of the more than 18 million college students enrolled in 2007 were 25 years old or older (NCES, 2009; Ross-Gordon, 2011). The projected higher education enrollment from 2007–2018 suggests that the number of students over 25 years will remain stable or increase during the current decade. Non-traditional students are defined as students who have enrolled in college at least one year after completing high school, have dependents, are a single parent, employed full-time, are financially independent, are attending college part-time, and/or do not have a high school diploma. According to this description, at least 73% of students could be viewed as non-traditional students. Adult learners enroll in higher education for both economic and personal benefits. The multiple roles of adult learners create challenges for the students who must allocate time for their personal commitments, academic work, and participation in campus-based organizations and activities.

According to Malcolm Knowles (1980), adults prefer self-direction in learning, have past experiences that can contribute to their learning, and have a readiness to learn based on a need to know something or do something. In addition, adults have a learning style that is problem-centered rather than subject-centered, and have a high degree of internal motivation. Non-traditional students, or adult learners, are a student population that is projected to increase. Higher education institutions should determine ways of teaching to accommodate this increasing student population. The design and delivery of programs are key to successful undergraduate experiences for these adult learners (Ross-Gordon, 2011).

Instructors typically follow one of two teaching methodologies—pedagogy or andragogy (Muduli et al., 2018). Knowles (1984) defined pedagogy as a teacher-led philosophy where the students are dependent learners and the teachers deliver class content instead of teaching students how to solve problems. With pedagogy, the teacher takes full control of the students' learning. The teacher is responsible for delivering the content of the class, creating assignments, tests, and exams, and determining the overall performance of the students. Conversely, andragogy is defined as a learner-centered teaching philosophy where students are independent learners, take control of learning, and accept responsibility for learning the material that they need. In andragogy, the teacher plays the role of a facilitator. The teacher focuses on solving problems rather than just delivering content. When the student is not mature and has very little knowledge about the subject, teachers usually prefer the pedagogical approach. However, when students are mature and are more familiar with the subject matter, teachers usually prefer the andragogical approach (Knowles, 1984). The adoption of a particular teaching methodology should be based on the learning needs of the students. (Muduli et al., 2018). The students' maturity is determined by the students' knowledge in the subject area, their interest in or need to learn the material, their willingness to accept the responsibility to learn, and the skills that the students already possess. The students' self-awareness is determined by the extent that the students take the initiative to determine their learning needs, define their learning goals, identify the resources that they need for learning, and implement and evaluate the best learning strategies for themselves. As people mature, they accumulate experiences that they use as a resource for learning. Learning motivation is determined by the students' willingness to learn the material as it is taught by the teacher. As a person matures, his or her learning objectives change from learning material that will be used in the future to learning material that will be used in the present. The students' learning also shifts from subject-centeredness to problem-centeredness. Adult learners select the courses that they want to take and prefer a learning style that will help them perform the tasks that they will encounter in their daily lives. Muduli et al. (2018) conducted a quantitative research study that included questionnaires that were randomly administered to 387 students who were enrolled in business schools offering management programs at the postgraduate level in two states (i.e., Gujarat and Rajasthan) in India. They received 313 questionnaires, which yielded an 81% response rate. The questionnaire included 24 items to assess the students' preferred learning styles and five items to assess the students' learning outcomes using a five-point Likert scale. Paired sample *t*-tests were used to determine if there was any overall significant perceived difference between andragogy and pedagogy and if there was a significant difference between students' perceptions of andragogy and pedagogy with respect to self-awareness, experience, motivation, and orientation to learning. The results of the study indicated that the students preferred andragogy significantly more often than pedagogy in respect to all four areas (Muduli et al., 2018).

Research on motivating students in colleges and universities has been conducted mostly at the undergraduate level. To determine the motivational factors for graduate students, Sogunro (2015) conducted an exploratory mixed methods research study with 203 adult learners, including 37 males and 166 females, who were enrolled in graduate degree programs. For the qualitative portion of the research, focus groups and interviews were used, and, for the quantitative portion of the research, a questionnaire was used. Sixty-three students participated in the focus group discussions, 115 students completed questionnaires, and 25 students participated in the interviews. The students' feelings of what motivated them to learn were recorded and grouped into 14 themes. Those 14 themes were condensed into eight themes. These themes were used to design the questionnaire. The responses for the questionnaire were analyzed using Excel to determine the means and standard deviation, then SPSS was used to conduct the paired samples *t*-test. The participants were asked to rank eight categories of motivating factors using a Likert-type scale. The following categories are listed in order of how the students ranked them from highest to lowest: 1) quality of instruction, 2) quality of curriculum, 3) relevance and pragmatism, 4) interactive classrooms and effective management practices, 5) progressive assessment and timely feedback, 6) self-directedness, 7) conducive learning environment, and 8) academic advising practices. Adult learning educators should know what these motivating factors are and how to implement them to help their students be successful (Sogunro, 2015). Although this study was conducted with graduate-level students, the motivating factors found in this research apply to undergraduate students who are adult learners.

To understand how adult students experienced and responded to a course specifically designed to use adult learning principles, Chen (2014) conducted a qualitative study that included 10 students from a psychology course. The study included seven female participants and three male participants. The average students' age was 45.4 years old. The group included one self-identified bi-racial student and nine Caucasian students. Chen sought to discover how the adult students in the course responded to self-directed and transformative learning principles. He also wanted to determine how effective these principles were in teaching the adults the course content. Interviews were recorded and transcribed. Five themes emerged from the interviews. The first theme of "reflection" helped make the material more meaningful to the students. The second theme of "emotional conflict" helped the students see that the material may be different from what they once believed. The third theme of "self-assessment" dealt with what the students thought that they knew. The fourth theme was crossing a "learning Rubicon" where the students were able to access where they were and where they were going. The final theme was the students actually making a "behavior change". By allowing students to move through these themes, to select their own topics, and to reflect upon each step, they made their learning more meaningful and increased their self-directedness. The research indicated that the adult students who had a voice in their studies were more vested in their education (Chen, 2014).

Adults have unique requirements for acquiring new knowledge and skills (Malik & Khaliq, 2017). Adult learning is very different from how children or even adolescents learn. Malik and Khaliq (2017) explored Knowles theory of andragogy to determine what characteristics in the medical field that adult learners found most important, what learning and teaching strategies were most successful for learning in adults, and how teachers could improve adult learning. They found that adults would evaluate the cost of their time and energy that was required to learn something new before devoting their time into learning it. To be effective, adults should be actively involved in the material that they are learning. Because adults have gained prior knowledge and experiences throughout their lives, they can use these experiences to build upon prior knowledge. Instructors can use these experiences to make the learning more meaningful to the adult student. When adults are ready to learn, they should be given the opportunity to learn immediately. Their time is valuable, and the instructors should consider their time. Adult education is focused on solving problems and learning strategies, not just learning subject material. Many times, adults can draw the information from their previous experiences and do not need to have the information repeated to them. Adults have both internal and external motivators. They learn because they want to learn or because it will improve their immediate situation. Using these strategies in the medical courses could provide a more successful method of learning that could lead to better retention of course knowledge and skills. Having a good learning experience will lead to an improved clinical experience and better care for the future patient of the student (Malik & Khaliq, 2017). These strategies could also be utilized in other lab courses, such as introduction to programming courses.

Historical Overview

The history of computing dates back to the creation of the abacus in 2700 B.C. (Ifrah, 2001). The abacus is believed to be the first device used for computing (Rau, Xie, Li, & Chen, 2016). In 1623, Wilhelm Schickard invented the first working mechanical calculator. In 1673, Gottfried Leibniz invented the digital mechanical calculator, the Stepped Reckoner, and was credited with documenting the binary number system, which is the number system still used by computers today (Ede & Cormack, 2017). Charles Babbage invented the Difference Engine No.1 in 1821, which was the first successful automatic calculator, earning him the title of the "father of computing" (Charles Babbage Institute, 2019).

Following these discoveries, the field of computing began to grow at an increasing rate. The history of computer science is categorized into generations (Doyle, 2015). The first generation of computers began in the mid-1940s during the Second World War due to the need for strategic types of calculations. The second generation also brought about the software industry by the creation of FORTRAN and COBOL programming languages. The third-generation computers became smaller due to the invention of smaller transistors that were placed on chips called semiconductors. The third generation was from 1964 to 1971 and included the addition of operating systems. The fourth generation lasted from 1971 into the 1980s. During this time, IBM introduced the personal computer, and more affordable versions of the personal computer were created. Following the 1980s, computer generations were no longer defined. Some scientist consider this period to be the fifth generation of computers. Computers began to use more than one processor, smaller devices that included touchscreen and voiceactivated features were developed, and the mobile computing field continued to grow. Along with these changes, applications that allowed easier and more sophisticated computer programs to be developed and applications written to be used on machines and for the internet began to grow rapidly (Doyle, 2015).

Computer science education has continued to grow since the 1940s. Columbia University in New York City offered one of the first computer science courses in 1946 (IBM, n.d.). The world's first computer science degree program began at the University of Cambridge in 1953, and the first computer science department in the United States was created in 1962 at Purdue University (Rosen & Rice, 1994).

Retention

Higher Education

Student retention has been a major issue for universities worldwide (Aljohani, 2016). When a university has higher rates of completion, the public tends to view that institution as having better academic, administrative, and financial statuses. Finding ways to improve student completion and retention has proven to be a difficult job. Aljohani (2016) reviewed student retention research to determine which items were commonly associated with student retention in higher education and which items were related most often to student attrition. In Aljohani's research, the student attrition factors were identified. The occurrence of each factor was counted and grouped based on their similarities into themes. The reasons that were given most often for student attrition were the quality of the students' experiences in the university and the level of student experiences with various academic and social systems, such as admissions, registration of courses, university policies, services available for the students, and the university facilities. The students' experiences in the university were the leading influence in their decision to stay at a university. According to Aljohani, much research has been conducted on why attrition happens, but not much research has been conducted on ways to move theory into practice (Aljohani, 2016).

Dong, Stupnisky, Obade, Gerszewski, and Ruthig (2015) conducted research to determine if causal attributions could predict students' success in college. Causal attributions are the explanations that are given for the students' outcome of the course. The researchers evaluated value as a mediator of the relationship between causal dimensions and academic success, as measured by perceived academic success and academic emotions. In their quantitative study, 389 students (i.e., 307 female and 82 male) from psychology courses at a medium-sized research university in the mid-western United States completed an online survey. Students were asked to answer the question, ''what cause might lead to your failure in this course?,'' by rating the cause that they listed (Dong et al., 2015, p. 535). Twelve additional items were listed in the survey with a nine-point scale for the students to rate other attributes. Three statements related to the students' feelings of causality, stability, external control, and personal control were included in the survey. The researchers found that causal attributions significantly predicted the academic success of students (Dong et al., 2015). When students made the decision themselves to be successful and were motivated to learn by themselves, their college education meant more to them, and they viewed their academic success in a more positive way. Because 91% of the students were Caucasian from a single university, the study may not be generalizable to other populations and universities. The study utilized a cross-sectional research design, which by design limited causal inferences. The students self-reported their academic success, which could have caused their responses to be biased. Future research could include diverse students, including different cultures and ethnicities. Dong et al. (2015) stated that future studies could measure the students' objective academic success, such as test grades and GPAs, and subjective academic achievement, such as the students' perception of success.

Understanding what makes adult learning successful is important for students, teachers, and industry (Kellenberg, Schmidt, &Werner, 2017). To explain the influence of self-determination, self-regulation, and reflection in the success of adult learning and to determine how these three factors influence adult learning, Kellenberg, Schmidt, and Werner (2017) conducted a qualitative study by performing a review of previously published literature covering the three theories. The attributes that contributed to successful adult learning programs were determined based on their research. A triadic model that illustrated how the three factors of self-determination, self-regulation, and reflective learning could be integrated into learning programs to increase adult learning was developed. The model covered self-determination questions, a self-regulation section that included questions to help the user form a plan, a way to monitor the plan, a way to evaluate the plan, and a self-reflection question. The benefit of using the triad was that it included all of the levels of the learning process (Kellenberg et al., 2017).

To provide an overview of student retention, the theoretical models for retention, and some practical research on retention that has been conducted in student affairs departments, Burke (2019) conducted a literature review covering articles that covered retention since 2010. Retention is hard to predict, and defining the areas that lead to student retention involves many different factors. Furthermore, student retention greatly affects the financial standing of the universities because tuition and fees are critical components of a university's income. The articles were reviewed based on three theoretical models (i.e., Spady's Undergraduate Dropout Process Model, Tinto's Institutional Departure Model, and Bean's Student Attrition Model). These models were grounded in sociology and dealt with the relationship between the student and the university. The research implied that the characteristics of the students and the interactions with the university, both academically and socially, had a big impact on the students' decision to stay or to withdraw from the university. The students' level of engagement while attending the university greatly influenced their decisions. Burke concluded that in order to increase retention universities should focus on the people working at their institutions as well as their academic programming. Properly training the staff, providing competitive salaries, and improving student services could lead to student retention. Universities should invest in programs to improve the student commitment to the university. Universities should also take into consideration the demographics of the students (Burke, 2019).

Attrition is an ongoing problem in enabling programs and in higher education overall (Willans & Seary, 2018). Enabling programs in universities are the programs that are used to help students who are not prepared academically to be successful in the university. The programs help the students to gain the knowledge, skills, and confidence that are needed to succeed in their program of study. The purpose of Willans and Seary's (2018) study was to determine the factor(s) that led students to withdraw from the enabling program and which areas were both common and different among the students who stayed in the program. The researchers' qualitative research included 23 students who enrolled in the program between 2013 and 2015 but did not complete the program. The 23 students included 13 females who were 19 to 64 years old and 10 males who were 19 to 67 years old. Interviews were conducted with the students. Students were given information about the study, and the students signed consent forms before the interviews were completed. The students participated in a semi-structured phone interview that was administered by an independent research assistant. The students were asked to tell why they withdrew from the program and what could have been conducted to encourage them to stay in the program. During interviews, 10 coordinators were asked to give reasons why students withdrew from the program and to give recommendations on how to reduce the number of students who withdrew from the program. The interviews were transcribed and digitized for analysis. Participants were de-identified and labeled as S for students and AC for access coordinators. The researchers reviewed the data and determined patterns and connections. Using the recordings and transcriptions, the researchers conducted data analysis to determine the common reasons for student withdrawal. The personal challenges that were identified included the students' physical and mental health, ability to juggle multiple roles, fear of failing, feeling of being connected to the staff, other students, and the institution, and feeling of not being connected to the institution. The institutional challenges that were identified included that students did not have enough academic support, particularly in their native language and a lack of technology available, such as internet speed and digital literacy. The coordinators' suggestions included for the students to be more committed to their studies. The students wanted individual support, uniform and understandable terminology across the program, and smaller class sizes. Both students and coordinators suggested better assessment and dedicated counseling services, and they recommended that assumptions should not be made about students' computer skills and accessibility. Willans and Seary (2018) concluded that more promotion of the university's support services was needed, which included having a dedicated counselor for students who needed additional support. They also concluded that having staff support for distance learning students during evenings and weekends could help relieve the fear of failure and the feeling of disconnect that were found. The researchers recommended that future research on these same issues could be conducted at other universities to determine if these factors contributed to attrition from those universities (Willans & Seary, 2018).

Introductory Computer Science Courses

Programming courses are considered gateway courses for students majoring in computer science undergraduate degree programs. These programming courses are reported to have low success rates, which lead to low retention rates in the computer science degree programs. According to a study by Beaubouef and Mason (2005) that was conducted in southeastern Louisiana, a university had over 400 declared computer science majors, but only 15 to 20 of those students graduated yearly. Most of the attrition happened during (or between) the freshman and sophomore years. The study found six main areas that contributed to the high attrition rate in those classes. These areas were poor advising, both before the students entered college as well as once they were in the program, a lack of math skills, poorly designed lab courses, a lack of practice using the concepts taught, the use of graduate teaching assistants who may have had a language barrier, particularly if students were international students, and poor time and project management (Beaubouef & Mason, 2005).

Failure and success rates of students in introductory programming courses were the focus of a study by Bennedsen and Caspersen (2007). At the time of the study, no worldwide statistics existed that documented the failure, withdrawal, or pass rates for these courses. Requests were sent to 575 institutions internationally. From the 80 responses received, only 63 responses were completed and included in the data analysis. Because the response rate was low (i.e., 12.7%), the generalizability of the study could be compromised. Each of these institutions responded by completing a questionnaire about their institution, teaching practices, and pass, fail, and withdraw information. The institutions that participated in the study varied in the type of programming instruction used and evaluation methods used. The results indicated that a high failure rate of around 33% was common among these institutions (Bennedsen & Caspersen, 2007).

After 7 years, Watson and Li (2014) continued the research that Bennedsen and Caspersen (2007) had begun. Watson and Li (2014) also reported an approximate 33% failure rate. Watson and Li based their study on 161 introduction to programming courses in 51 institutions located in 15 countries. Despite differences in the samples for Bennedsen and Caspersen (2007) and Watson and Li (2014), the pass rates were similar. According to both of these studies, although the pass rate for introduction to programming classes was around 65%, the average pass rate for the preceding three terms was 68%, which was comparable to worldwide results. The average withdrawal rate for the three terms was 26%, which was normal for these classes but high compared to most of the other courses that were taught in computer science majors (Bennedsen & Caspersen, 2007; Watson & Li, 2014).

Bennedsen and Caspersen (2019) revisited their original research 12 years later to evaluate if the success rate had changed over the years. Because their original research had a low response rate (i.e., 12.7%), Bennedsen and Caspersen recommended that the Association for Computing Machinery Education Council help to provide reliable data to validate their study. The recommendation was not implemented to their knowledge, so Bennedsen and Caspersen replicated the study. They used the same criteria that they had originally used to determine whom to send the data requests. For the study, 1,020 requests were sent out, and 170 (17.8%) were returned. The results of the study indicated that the pass rate had only increased 5% from the original study. The pass rate went from the initial 67% to 72% in 12 years (Bennedsen & Caspersen, 2019).

Various research studies have examined the causes of such high failure rates. Hawi (2010) conducted a survey with 45 students who had finished an introduction to programming course. The students represented different genders, cultural backgrounds, and levels of achievement. The students gave 10 reasons that they felt were a possible reason for failure in the course, which included not enough time, not enough time spent studying, not having a learning strategy, not practicing the assignments, not putting in enough effort, inappropriate teaching strategies, subject difficulty, exam anxiety, cheating, and unfair treatment. The learning strategy in the classroom was the main reason given by the students for failure in the study (Hawi, 2010).

Bosse and Gerosa (2017) conducted research to determine why programming was difficult to learn. The aim of their research was to identify areas that caused students difficulties when learning how to write computer programs. In their mixed methods research completed at the University of São Paulo, the enrollment and final grades for

18,784 students who registered for introduction to programming between 2010 and 2014 were collected. The researchers also collected journals from 34 students from six courses in 2015 that the students completed about their time taking the course. Fourteen instructors of the introduction to programming course at the University of São Paulo were selected randomly and were interviewed. The number of student registrations were counted, and the pass, failure, and withdrawal rates were calculated for the students that registered for the course. Journals from the students in the study were analyzed using grounded theory procedures and then grouped by the four categories that emerged, which included difficulties, study strategies, preferences, and self-assessments. Instructors of the course were interviewed, and the interviews were transcribed and coded to find suggestions for improving how the courses were taught. Of the 18,784 students who registered for the course during the evaluation period, 30% of those students either failed or withdrew each year. One of the areas where the students experienced difficulty was working with functions. Students also struggled with understanding the scope of the variables and passing and returning parameters. The instructors agreed that the main area where students struggled was logical reasoning. In addition, the instructors stated that students had trouble with understanding arithmetic, logical, and relations operators, loop structures, array indexes, and the scope of the variable in functions. Other factors that made the class difficult to teach were the makeup of the class, the size of the class, the students who were not interested in learning, and the trauma of the students who had to repeat the course. Determining the areas that made learning to program difficult for students helped the instructors prepare their lessons better and helped to develop new ways to teach programming, which helped the students learn the material better (Bosse & Gerosa, 2017).

Vihavainen et al. (2014) conducted a quantitative study to determine the impact that different teaching practices have had on pass rates for introduction to programming courses. The research questions that the researchers sought to answer were "1) How do teaching interventions reported in the literature increase students' success in CS1 [introduction to computer programming]?, 2) What practices do the successful teaching interventions comprise of?, and 3) Do so called best practices, or practices that are significantly better than others exist?" (Vihavainen et al., 2014, p. 20). The quantitative systematic review was based on 32 published articles covering the impact of different teaching practices in introduction to programming courses. The articles were selected based on the fact that the articles contained keywords, such as improve, increase, decrease, lower and retention, attrition, pass, fail, success and programming, introductory programming, and CS1. Databases, such as ACM, IEEE, and Google Scholar, were searched to find the relevant articles. Details, such as the course details, year, semester, teaching practices, totals, and percentages (i.e., pass, fail, withdraw), were collected and counted, and the teaching interventions were coded and analyzed from three different viewpoints. The teaching interventions that included group work and collaboration and creating a course to prepare the students for the introduction to programming course were the highest performing activities. These interventions increased the pass rate by an average of one-third. A large percentage also reported increased student/teacher collaborations and updated teaching materials and content. However, only successful practices were reported. Knowing what was not successful could help teachers avoid using these strategies. Many limitations of the study were given. The teaching approaches were not discussed in detail but assumed to be a traditional lecture and lab-based approach. The learning objectives were not considered, and the quality of the teaching intervention design and experiments were not considered. Also, the final number of selected articles (*n* = 32) was low, which could have been due to selective reporting. The effect of the teacher and the effect of different student populations among different institutions were not considered and the results were based on the terms used for tagging the articles and may be different for other classification approaches. In addition, the definitions for "pass rate" were not consistent in all articles (Vihavainen et al., 2014).

A lecture and lab-based instructional strategy has traditionally been used to teach introduction to programming courses (Canedo et al., 2018). The primary instructional strategy utilized within a course could serve as a reason for its low success rates. After many years of increasing failure rates in the beginning computer science courses, Canedo et al. (2018) developed a questionnaire available to all of the students at the University at Brasilia that had taken the introduction to computer science, basic computing, and algorithms and computer programming courses. The questionnaire provided a space for the students to give their personal opinions for a qualitative analysis, and 637 students completed the questionnaire. The students felt that the number of students in the classes was too large and did not allow for enough individual attention from the professor. The students stated that the computer labs contained many broken computers. Another negative factor that the students stated was that the professors needed to have better material, more exercises, and solutions for those exercises. The students determined six areas that they thought were good qualities of University of Brasilia. The areas included the use of teaching assistants in the courses, the feeling that introduction to computer programming was important to their degree program, the professors used a set instructional strategy, the professors had leadership skills, they enjoyed working on research projects with the professor of the class, and the programming language that was being taught was helpful in learning the subject (Canedo et al., 2018).

At Al-imam Muhammad Ibn Saud Islamic University, Alturki (2016) noted that as many as 65% of the students in introduction to computer programming failed or withdrew from the course. Because the course was a prerequisite course for all of the other computer science courses, when students failed or withdrew from the course, the students fell behind in their course work, which delayed their graduation. Previous research had shown that the students' failure could be related to the students' motivation and the instructional strategy used to teach the class. Students were evaluated on various graded assessments, such as quizzes, assignments, attendance, tests, midterm tests, and final tests. In Alturki's research, he wanted to determine if the grade distribution for these assessments was the best way to evaluate the performance of the student properly and to help increase the students' motivation in the course. The individual grades of 138 students during four terms with different instructors and sections were analyzed. To analyze the individual grades, the grades were converted to percentages. The means and standard deviations were calculated. Quizzes and labs had the highest variation in grades. The quizzes and labs were completed immediately after covering the material, which could be the reason that the students scored higher on these assignments compared to the midterm and final exams. The midterm had the least variation and was determined to be the best indicator of the final grade. Students who had zeros on lab assignments sometimes performed well on exams and the final exam and passed the course. These zero scores possibly indicated a low motivation to complete the assignments and were not a good indicator of the overall grade for the course. Lab assignments allow the students an opportunity to apply their knowledge and skills, but they should not be given a high percentage scale for the final grade. A simpler method of assessment with fewer assignments would allow the students to focus on the main goal of the course and improve their motivation (Alturki, 2016).

One of the causes of low success rates that has been researched is the different instructional strategies used to teach the introductory programming courses. A study that was conducted by Sarpong, Arthur, and Amoako (2013) to determine the causes of student failures in the class. The study involved 100 students at Valley View University in Adentan-Accra, Ghana, who were taking first-year programming classes. In their research, Sarpong et al. found that 82% of the participants thought that problem-based teaching was the most appropriate way to teach these courses. The next method that 79% of the participants believed would be the most effective was pair or group learning. They noted that the instructor should use these techniques to promote teamwork and to increase the participation of all of the members of the group (Sarpong et al., 2013).

Hegazi and Alhawarat (2015) researched why the success rate in introduction to programming courses was low and how it could be increased. They explored instructional problems involving introduction to programming courses at Prince Sattam Bin Abdulaziz University in Kingdom of Saudi Arabia by interviewing the teachers who have taught the course and by having 239 students complete a questionnaire. Hegazi and Alhawarat analyzed the results of the interviews and questionnaires, and solutions were proposed based on their research and previous research. They found that the average pass rate for the three years was 68%. This pass rate was consistent with the pass rate at other universities worldwide. The withdrawal rate during the period of the study was 26%, which was considered high compared to the other courses taught at the university. One of the reasons for the high withdrawal rate was that, when students failed, they told other students, which disseminated the thought that the course was very difficult. Therefore, when new students came into the class, they had the preconceived notion that the course was very difficult. The students gave a score of 3.3 out of a possible 5 for their overall satisfaction with the course. The students felt that teaching the entire course as a lab course would be more beneficial than using the traditional lecture and lab format. Only a few students thought that exams were a good measurement of their performance in the course. They also believed that more problem-solving elements should be added to the course. By determining the students' and the instructors' views of the course, future courses can be developed to take advantage of these findings (Hegazi & Alhawarat, 2015).

Horton and Craig (2015) explored groups of students who took introduction to programming courses using a traditional lecture-based instructional strategy and students who took courses that were taught using an inverted instructional strategy. They also sought to determine which characteristics had an influence on whether the students withdrew from the course, passed the course, or failed the course. They conducted a quantitative research over two courses, which included 1,236 students. The first course was taught in Fall 2012 using a traditional lecture-based instructional strategy that included 542 students. The second course was taught in Fall 2013 using an inverted-type instructional strategy that included 694 students. Information was collected regarding prior experience, English fluency, initial enthusiasm in taking the course, and reasons for taking the course. Multiple regression was used to examine the relationship between exam scores, prior experience, and the type of course taken (i.e., traditional or inverted). The results showed that students who took the introduction to programming course as a requirement of their major tried harder in the course. The exam average was higher in the inverted course regardless of why the students were taking the course. This higher exam average could have occurred because the students who were in the inverted course were more likely to withdraw from the course early on due to their lack of problem-solving skills. In the traditional course, only 28% of the students who failed the midterm exam were able to pass the course successfully; however, in the inverted course, 55% of the students who failed the midterm course were able to pass the course successfully. The researchers were not able to determine how much the students who withdrew from the course affected the increased exam scores that were reported. Their research indicated that the students in the inverted course had a better chance of passing the course regardless of whether or not they failed the midterm (Horton & Craig, 2015).

The students' experience when they take their first computer science course can determine if the students stay in the major or change their major. Traditionally, a computer programming course is the first course that computer science majors take. Peterson, Craig, Campbell, and Tafliovich (2016) conducted a qualitative study to determine the reasons that students decided to withdraw from the course. They conducted interviews with 18 students in an introduction to programming course at a large research university in North America. The students were from separate campuses of the university. The interviews were coded and compared with the other researchers' codes until the researchers agreed upon a final set of codes. The perceived reasons for withdrawing included the introduction to programming course required too much work, ineffective study habits, prior programming experience was needed, students relied on someone else to complete their work for them, and the thought that withdrawing from the course was acceptable and the course may take multiple times to pass. The researchers concluded that success in the course could be improved by providing support during the term, teaching and demonstrating good study habits, creating assignments that are at the level of the student, and providing effective feedback (Peterson et al., 2016).

The number of students who fail to advance in the computer science degree has been a concern for institutions in the western world (Quille & Bergin, 2019). The students' failure to complete the introduction to programming course successfully is one of the main reasons for the high attrition rate in a computer science degree program. An attrition rate of 25% in the computer science major was reported in Ireland compared to the national average of 16% for other fields of study. Quille and Bergin (2019) conducted their research at 10 institutions in Ireland and one institution from Denmark, which included 692 complete student data sets. The purpose of their research was to examine improvements within the classrooms. Student demographics and grades were collected four to six hours into the introduction to programming course. A survey was used to collect additional information, such as programming self-efficacy, mathematical ability, and the number of hours that the students spent playing video games. The intervention consisted of the lecturer telling personal experiences and giving testimonials from other students, presenting current research (e.g., brain activity), and constructive feedback on the process used by the students. Scratch programming was taught at the same time that the introduction to programming course was taught. A Welch's *t*-test and a one-way AVOVA were used to analyze the data before and after the intervention. Scratch programming was recommended to be taught at the same time as the original programming module. The 2015-2016 cohort had an average grade of 66.71%. The intervention group in 2016-2017 had an average grade of 75.39%. This increase in the average grade indicated that the intervention was successful (Quille & Bergin, 2019).

To learn how the students' emotions were influenced by different learning designs and what the students' internal motivation was for learning material, Lykke, Coto, Jantzen, Mora, and Vandel (2015) completed a mixed methods experimental controlled comparison study with first-year students in introduction to programming courses at the Informatics School, Universidad Nacional de Costa Rica. The university had a high average failure rate of 47.2% in the introduction to programming courses between 2008 and 2012, and motivating students was an ongoing problem. Their study included 15 groups of students. The students included five groups per learning design. The control and problem-based groups had a maximum of 25 students, and the problem-based design combined with LEGO Mindstorms Robots group had a maximum of 20 students. Lykke et al. used a survey, interviews, observations, and focus groups in their research. For the survey, 229 students responded. The survey was used to measure the students' understanding and attitudes concerning the learning design. Informal interviews and observations were conducted with six groups. Six classroom observations were conducted to document the students' behaviors and emotions in the classroom. Three focus group interviews with 10 students from each learning design were conducted to validate the findings from the initial three studies. The results indicated that the learning design influenced the students' emotions and engagement in the course. The students thought that working with the robots was fun and interesting, but they were frustrated by the amount of detail that was needed. The problem-based learning students liked the interaction between the professors and the groups while completing assignments. However, they found that working together could be challenging. The control group was very passive and lacked interaction with the professor and students. Although they felt secure in the traditional format, they were not secure in their ability to complete the assignments. The researchers concluded that all three designs had their advantages and disadvantages. None of the designs was perfect by themselves. The robots design particularly would need the students to have more theoretical knowledge about programming, and the assignments would need to be improved so that the robots design was more beneficial to the course. A limitation of the study was the evaluations being based on the students' emotions, which were subjective, meaning that the students' emotions relied on the students' mood, their current knowledge, and their current interests. The emotions were also context-dependent, meaning they depended on the surroundings, such as the weather, noises, and the current social issues. In addition, the emotions were dynamic because their experiences were constantly changing (Lykke et al., 2015).

Summary

Beaubouef and Mason (2005) found withdrawal rates in introduction to programming courses were reported to be as high as 40% at many universities. The course design along with not enough practice using the concepts being taught and poor time and project management were found to be some of the reasons for this high withdrawal rate (Beaubouef & Mason, 2005). Because there was no documented worldwide statistic for the failure, withdrawal, or pass rates for introduction to programming courses in universities, Bennedsen and Caspersen (2007) conducted their own research and determined that an average failure rate of 33% was considered the norm for these courses. Seven years later, Watson and Li (2014) conducted their own research and found that the failure rate was still around 33%. Bennedsen and Caspersen (2019) replicated their previous study. This study included an increase of 170% responses but only showed a 5% decrease in the failure rate of the original study.

According to Hawi's (2010) research, the learning strategy utilized within the classroom was the main reason given by the students for failing the course. Sarpong et al. (2013) found that the students did not have enough problem-solving skills, analytical thinking skills, logical skills, and planning skills. Seventy-nine percent of the participants believed that pair or group learning would promote teamwork and increase participation to help the students learn these concepts and improve the success rate in the course (Sarpong et al., 2013). Horton and Craig (2015) also found that students could have a better chance of passing the course if they took the course using an inverted delivery instructional strategy. Figure 1 presents a concept analysis chart for studies related to retention.

Study	Purpose	Participants	Design/ Analysis	Outcomes
Beaubouef & Mason (2005)	To identify causes of high attrition rates in computer science and information technology undergraduate degree majors.	400 computer science majors	Qualitative Survey	• The causes of attrition included poorly designed courses, lack of practice and feedback, and poor management skills.
Bennedsen & Caspersen (2007)	To determine an official statistic to prove the claims that introduction to computer programming courses are known for their high failure rates.	63 international institutions	Quantitative Survey	• The findings were an average failure rate of 33% was considered the norm for these courses.
Bennedsen & Caspersen (2019)	To follow up on their previous research because no other research has been conducted to provide data from more institutions.	170 international institutions	Quantitative Survey	• The pass rate increased 5% from the initial 67% to 72%.
Hawi (2010)	To determine the causes of success and failure in introduction to programming courses.	45 computer science students	Qualitative: HyperRESEARCH coding and evaluation	• The causes of failure included the learning strategy used, a lack of studying, a lack of practice, and

Study	Purpose	Participants	Design/ Analysis	Outcomes
				the teaching method used.
Horton & Craig (2015)	To determine what influences students to withdraw, fail, pass, and/or take advanced programming afterwards.	1236 students in introduction to programming courses	Quantitative: multiple regression to examine the relationship between exam grades, prior experience, and type of instruction.	• Students taking introduction to programming as a major requirement tended to try harder in the course. • The exam average was higher in the inverted course.
Sarpong et al. (2013)	To determine the causes for failure in introduction to programming courses.	100 students in introduction to programming courses	Quantitative Survey: using SPSS	• The reasons for failure included lack of problem-solving skills, lack of planning, and lack of feedback.
Watson & Li (2014)	To provide additional evidence of a high failure rate in introduction to computer programming courses.	161 introduction to computer programming courses in 15 different countries in 51 institutions	A review of literature and a statistical analysis of the data on pass rates from the articles.	• An international pass rate of 67.7% was found.

Figure 1. Concept Analysis Chart for studies related to retention.

Team-Based Learning

Computer Science

According to Makalew and Pardamean (2017), traditional lecture and lab-based learning is textbook-centered where the students follow the instructions given by the instructor with no room for their own creativity. In traditional lecture and lab-based learning, the students are considered to have no prior knowledge of the subject and are given the information that they need by the instructor. Traditional lecture and lab-based learning is considered the simplest strategy of teaching information and requires the least amount of preparation. Using technology, such as PowerPoint presentations to supplement lectures, the traditional lecture and lab-based learning strategy has been improved slightly. Combining technology with lectures can increase engagement in the classroom. With team-based learning, the learning process is different compared to traditional lecture and lab-based learning. With team-based learning, teams should remain the same throughout the entire course, the grades should be given immediately so that the students know where they need to improve, and feedback should be immediate and specific. When using team-based learning, one of the goals is application of the learned content to solve problems together as a team (Makalew & Pardamean, 2017).

Makalew and Pardamean (2017) conducted research to measure students' motivation, engagement, and academic achievement in a computer science course when changing the instructional strategy to a team-based learning format. Their research consisted of 64 participants who were taking a mobile game creative design class at Bina Nusantara University in Jakarta, Indonesia. For the study, motivation was measured using the Motivation Strategies for Learning Questionnaire using a seven-point Likert-type scale. The questionnaire measured the students' motivation and the effect on learning strategies based on the instructional intervention. The questionnaire covered six areas of engagement. Academic achievement was measured by questions that were developed by the instructor based on the material taught in the course. The Shapiro-Wilk normality test was applied with an alpha level of .05 to insure normality of the data. The findings indicated that the motivation and engagement levels were not affected, but the academic achievement level had a significant improvement with a p value of .0000271. Makalew and Pardamean recommended that more research could be conducted on incorporating more technology into the development of team-based learning courses to make them more effective.

To determine the students' feelings about using team-based learning in a computer science course, Kirkpatrick (2017) conducted an explanatory mixed methods observational study. The computer organization course was taught during Spring 2015 at a university in the eastern United States. The course was taught in four different sections. Two sections were taught using team-based learning, including 28 students in one section and 27 students in the other section. Two sections were taught using a traditional lecture format, including 27 students in one section and 29 students in the other section. The quantitative study consisted of a pre- and post-survey using the validated Value of Teams Survey. The qualitative study consisted of two semi-structured focus groups that were conducted by a separate faculty member to reduce the chance of instructor bias. Identifying information was removed from the transcripts before they were analyzed. The two focus groups, which included one member from each team, were conducted during the 12th week of class and were led by a faculty member from the College of

Education. The sessions were recorded and transcribed anonymously by a professional transcription service. The transcripts were analyzed using open and axial coding techniques for thematic interpretation. Only the qualitative portion of the study was reported in the study. The researchers stated that the other results would be reported in a later paper. The researchers used the student focus group comments to determine the strengths and possible problems of implementing team-based learning in computer science. Five common themes were found. The most common theme was the value of in-class collaboration. The students felt that using team-based learning allowed them to discuss the material with their teammates and to ask more questions. The students stated that they perceived to learn more material but also stated that the preparation that had to be completed prior to coming to class was more difficult compared to the traditional course. The second most popular theme was creating the teams was critical but also had problems. The students did not agree on the best method for selecting the teams. The third theme was the factors that made a successful team. Most students agreed that a team member's effort counted for more than his or her academic ability. The students also agreed that it took time to work together as a team and build a strong team. The students liked the peer evaluations, but they felt that the evaluations could be completed throughout the course and not only at the end of the course to help members see where to improve. The fourth theme was that the readiness assurance process was difficult. The students stated that the material was difficult to read without being explained by the instructor. The final theme that emerged was that completing the team exams was time consuming due to student discussions and difficult to determine the best plan to complete the exam. Most students perceived that team-based learning was a good use of the time in the class and that the discussions in the class helped them to learn the material better. Finally, in order to triangulate the findings of these two studies, the qualitative codes and themes were mapped back to the individual Value of Teams Survey questions (Kirkpatrick, 2017).

Many instructional strategies have been used over the years to teach introduction to programming courses, including team-based learning. Within the last 10 years, research and analysis has been conducted to compare team-based learning to the other instructional strategies used. Lasserre and Szostak (2011) evaluated the effect of team-based learning on the two largest concerns for computer science instructors, which are increasing retention rates and success rates of the students in introduction to programming courses. Lasserre and Szostak evaluated four years of introduction to programming courses from the University of British Columbia Okanagan, which included 294 students. During the first year, the course was taught using a traditional lecture format, and, during the following three years, the course was taught using the team-based learning approach. The results reflected major improvements in the withdrawal rate and the success rate of the students who completed the course. The success of the students was measured by the final exam scores. They found that the number of students who made a score of 50% or greater on the final exam increased their score from 54% to 75.5%. They also found that the withdrawal rate went down from more than 30% to 6.4%. The greatest benefit of using team-based learning to teach introduction to programming courses was the increased programming skills that it provided for the students and the increase in the students' confidence in their ability to program (Lasserre & Szostak, 2011).

Matalonga, Mousqués, and Bia (2017) conducted a qualitative study based on two team-based learning courses that they designed, which were taught at Universidad ORT Uruguay. The researchers evaluated the outcome of the course compared to published results from similar studies. In order to compare their results with previous results from Michaelsen et al. (2004), the same survey was administered. The researchers included an additional seven questions to the survey, which evaluated the specific course and determined if any improvements could be made to the course. The researchers sent a survey to the students who were enrolled in the two courses in order to determine how the students felt about taking the course using team-based learning. The first course, agile software engineering, included 63 students, and the second course, software architecture, included 22 students. Only 21 completed responses (i.e., 15 from the agile software engineering and 6 from the software architecture) were received from the 85 students. Most of the responses from the survey administered by Matalonga et al. (2017) were similar to the responses from the study conducted by Michaelsen et al. (2004). The students thought that a course taught using team-based learning had more advantages than a course taught using a traditional lecture-based format. The students also believed that team-based learning allowed them to use their own individual learning style. A part of the survey revealed a large difference between Michaelsen et al. (2004) and Matalonga et al. (2017). In Michaelsen et al. (2004), 23% of the students felt that team-based learning could not be implemented in other courses compared to 47% in Matalonga et al. (2017).

Team-based learning has been found to be an effective teaching methodology. However, without guidance and the necessary tools, team-based learning can fail. Wang and Hwang (2017) conducted a study to assist in the development of a team-based learning activity in a sophomore level computer programming course where teams of students created programming problems that were solved by other teams in the course. They conducted a quasi-experimental quantitative study, which included 53 sophomore students from two classes at a university in eastern China. The treatment group included 25 students, and the control group included 28 students. The treatment group was taught using the problem posing teaching strategy. Students took a pretest to determine their programming skills and a pre-questionnaire using a Likert-type scale to determine their self-efficacy for team-based learning. The same instructor taught the students for 7 weeks. After the 7 weeks, another pretest was administered over the material to determine how much the students had learned, and another pre-questionnaire was administered to determine their self-efficacy for team learning and cognitive load. For the next five weeks, the students were divided into the control group and the treatment group. During this time, the students in the treatment group were divided into teams and completed programming assignments that were created by the other teams. Following the five weeks, the students were given a posttest to determine how much they had learned and a post-questionnaire using a Likert-type scale to determine their self-efficacy for team learning. An ANCOVA was used to determine if significant differences existed between the treatment and control groups. The results indicated that the problem posing-based practicing strategy could significantly improve the amount of material that was learned by the students. The results also indicated that using this teaching strategy could lower the students' cognitive load. The study had a few limitations. The sample size was not large, so the findings cannot be generalized to large numbers of students. The proposed approach was only used in the course for 7 weeks, so the findings cannot be generalized to all of the sections of the course during an entire term. The effect sizes for the statistical results of cognitive load and self-efficacy were not large, which suggested generalizability of these findings were limited. Another limitation of the study was the assumption that the students had the same programming knowledge for creating problems as they did for reviewing the problems after the other teams completed them. For future research, long-term experiments with larger sample sizes were needed. These strategies could be used when teaching all of the units in the course to determine the impact on all of the units as a whole (Wang & Hwang, 2017).

Student Attitudes, Perceptions, and Motivation

Educators realize that using the traditional lecture-based teaching style might not be the best way to meet their learning outcomes. Remington et al. (2017) conducted their research in an elective pharmacotherapeutics course to compare students' attitudes and perceptions about team-based learning compared to lecture-based learning. They conducted a qualitative study, which included 30 students in two different classes. Their study included test grades, a survey that included Likert-type scale questions and open-ended questions, and a written reflection from the students. All of the students in the course participated in the study. A repetitive coding process was used to make certain that the codes, categories, and themes used represented the responses that were given by the students. The results were discussed with two other researchers to prevent researcher bias. Quotes from the students were used to support the themes that were selected. Students stated that they liked the application exercises and team discussions in team-based learning but did not like the work that they had to do on their own to prepare for the course. For the lecture-based course, the students stated that they liked that they did not have to prepare for the class in advance but did not like that application exercises that covered the material that they were learning in the lecture were not included in the course. The students stated that they were able to retain the material better using team-based learning. The students perceived that they were more engaged in the course and learned more from the insight and discussions with their peers. Using team-based learning improved the test scores of the students as compared to lecture-based course. The students also perceived that they had more confidence in their ability to make decisions after taking the course using team-based learning. Remington et al.'s research supports using team-based learning, which can be beneficial in a clinical setting. The researchers found a difference between what students claimed they liked in an instructional strategy and what actually helped the students to learn. Limitations of their study included that the study was completed at one location with students who had been taught using team-based learning previously. The small sample size was also a limitation of the study. Another limitation was the course was an elective course, and the students knew before they elected to take the course that it would be taught using team-based learning, which may have limited the number of students who did not like team-based learning from taking the course (Remington et al., 2017).

Frame et al. (2015) conducted a quantitative study to evaluate student perceptions of team-based learning compared to a traditional lecture-based learning format in pharmacy courses. The researchers evaluated 111 first-year pharmacy students at two different universities during two different semesters. The students were divided into teams of five to six students with each group containing at least two males. The team members were selected based on their personality type. The students in the study completed a 22-item team perceptions instrument before and after the fall semester. A 14-item teaching style preference instrument was completed at the end of the following spring semester. Pre/post changes on the perceptions of using team-based learning were calculated using the Wilcoxon signed rank test, and the differences between the two universities were calculated using the Mann-Whitney U test. The differences between the data for the universities using team-based learning and the universities using traditional lecture learning were determined using the Mann-Whitney U test. Finally, comparisons were made on the seven themes (i.e., thinking critically, problem solving, being more prepared for examinations and quizzes, keeping up-to-date with the material, and pedagogy preference) between team-based learning and traditional lecture-based using the Wilcoxon signed rank test. When team-based learning was used the first semester, students developed a more positive view of teams and teamwork by the end of the semester than students who were taught using traditional lecture-based learning. Students, who had two team-based learning courses before taking a lecture-based course, preferred the team-based course, but the students, who had a lecture-based course before taking a team-based course, preferred the lecture-based course. The research indicated that taking a team-based learning course influenced the students' view of how useful the teaching strategy was. The researchers suggested that universities should implement team-based learning early in the curriculum to allow students time to understand the benefits that team-based learning added to their education. In addition, when creating teams, students with different views, perspectives, and strengths should be included on each team (Frame et al., 2015).

Jeno et al. (2017) conducted research to determine the effects of team-based learning on motivation and learning. The researchers investigated whether the implementation of team-based learning, compared with lecture-based learning, had an influence on the students' motivation to learn, their perceived competence, their perception of the instructor as supportive, their level of engagement, and their perceived level of learning. They conducted quasi-experimental pretest-posttest research with biology students from a large university in Norway. The research group consisted of 24 students, which included 11 males (45.8%) and 13 females (54.2%). The participants used a seven-point scale to respond to items that measured their intrinsic motivation, identified regulation, external regulation, amotivation, competence, needs satisfaction, autonomy support, engagement, and perceived learning. Following a two-week period of traditional lecture-based instruction, the students completed the pretest questionnaire. For the next four weeks, the students continued being taught using traditional lecture-based instruction. Next, the students were taught for a two-week period using team-based learning. At the completion of the team-based learning instruction, the students completed the same questionnaire again. Repeated sample *t*-tests were conducted to determine the change in scores between the pretest and posttest. A path analysis was conducted to determine how well the self-determination theory fit together and to assess the indirect effects. The researchers concluded that using team-based learning added a new and exciting element to learning, which increased the students' intrinsic motivation and external regulation. The students' competence also improved after introducing team-based learning. The students' improved competence was due to the use of small discussion groups, immediate feedback from the readiness assurance tests, and the teachers' increased guidance and support. Jeno et al. recommended that teachers include active-learning approaches, specifically team-based learning, into their teaching. Their results indicated that students perceived the instructors as more autonomy supportive when using team-based learning. The use of team-based learning also improved rote learning and conceptual learning (Jeno et al., 2017).

Comparison of Team-based Learning with Other Instructional Strategies

Rezaee, Moadeb, and Shokrpour (2016) conducted a quasi-experimental research to compare the outcomes of teaching team-based learning and traditional lecture-based learning in a hospital management course. The study included 25 undergraduate students in the Management School of Shiraz University of Medical Sciences. The courses were divided into two parts. The first half of the course was taught using interactive lectures, and the second half of the course was taught using team-based learning. The students were assessed before the beginning of the course and two months after the course was completed to determine how much knowledge that they had regarding the course material. The mean of the final exam scores was calculated, and the students' level of satisfaction about the teaching method was calculated. The students' satisfaction of using team-based learning was measured using a questionnaire that contained 17 items. The items on the questionnaire used a three-point Likert-type scale. The scores of the final test in the section that was taught using team-based learning were compared against the final scores of the section that was taught using the interactive lectures. The test scores of the students after completing the team-based learning section were higher than the test scores after completing the lecture-based sections ($p < .001$). The students also demonstrated a higher retention of the knowledge that they learned using team-based learning ($p < .001$). However, there was not a significant relationship found between the mean scores of the final exam in the team-based learning section and the interactive lecture section ($p = .116$). Most of the students reported that they preferred team-based learning to the interactive lecture-based course ($p = .037$). A limitation of the study was that the researchers were not able to divide the students into two separate groups where one group was taught using only team-based learning and one group was taught using only lecture-based learning (Rezaee et al., 2016).

The trend toward lower grades in science and engineering influences students' decisions to change their majors (Comeford, 2016). In order to improve the success rate in Comeford's introduction to chemistry class, he conducted a quantitative study that included five sections of introductory to chemistry prior to implementing team-based learning, which included 192 students, and nine sections of the course after implementing team-based learning, which included 285 students. Final grades were collected in the introduction to chemistry class for 14 sections between Fall 2008 and Spring 2013. A comparison of the number of students who withdrew, failed, or made a final grade of D in the courses before and after team-based learning was completed. Comeford (2016) determined that the failure rate in the course before implementing team-based learning was 31% and after implementing team-based learning was 19%. The results indicated that using team-based learning improved the success rate; however, the study did not determine why this improvement occurred. A limitation of the study was that the same instructor taught all of the courses. In order to test the generalizability of the study, more research would needed to determine if the same results would be produced if different instructors taught the courses (Comeford, 2016).

Internationally, medical schools have historically used problem-based learning to teach their courses. Over time, this instructional strategy has become less effective and led to less accountability by the students (Burgess et al., 2017). Unlike problem-based learning that requires many tutors for each class, team-based learning has the benefit of small group instruction and learning. The purpose of the study by Burgess et al. (2017) was to compare the students' perceptions of taking the course using a team-based learning approach verses taking the course using a problem-based learning approach. The mixed methods study included 169 first-year students at Sydney Medical School. Questionnaires containing open-ended and closed-ended items were given to students after they completed three sessions of a problem-based learning course and three sessions of a team-based learning course. The quantitative data that the students provided were analyzed using descriptive statistics. The qualitative data from the open-ended questions were coded and organized into the proper themes. Of the participants, 147 students completed the questionnaire after problem-based learning, and 152 students completed the questionnaire after team-based learning. The students reported that, when using the team-based learning approach, they liked the smaller size of the groups, the readiness assurance tests that were given, the immediate feedback that they were given, and the time efficiency of the team-based learning approach. The students reported that, when using the problem-based learning approach, the tutors were not consistent in their level of expertise, they were not given clear directions, and the large group size made learning the material difficult. The study provided reliable evidence that changing from using a problem-based learning approach to using a team-based learning approach could be beneficial to the students (Burgess et al., 2017).

Implementation

To evaluate how to implement team-based learning and to determine its effectiveness, Walker and Zheng (2017) conducted a mixed methods research study in a Singapore teaching institute. Their study included 30 teachers, between the ages of 25 to 59, with the majority between 30 and 39 years old. The participants were grouped into teams of five or six students based on the subject area that they taught, their gender, and their ethnicity. The quantitative data for the study were collected using a questionnaire that asked the participants' demographic information and closed-ended questions that asked the participants about their experience using team-based learning using a five-point Likert-type scale. The qualitative data were collected using a focus group discussion where one member of each team participated. The individual readiness test and the team readiness test scores for the participants were averaged, and a paired samples *t*-test was used to compare the means. The focus group discussions were transcribed, and the statements were grouped into four themes. The four themes were "most interesting", which was the exchanging of ideas in discussions, "most helpful", which was learning the applications and team readiness discussions, "peer ranking", which the participants disliked the most, and "punctuality", which the participants also disliked. Because the participants were part-time students and had other obligations, they were not always able to be in class on time, which caused them sometimes to miss the individual readiness test quiz. A bivariate Pearson's product-moment correlation coefficient was conducted, and a strong relationship was found between the individual readiness test and the final grade in the course. The students felt that they learned better using team-based learning and would prefer team-based learning to the traditional lecture-based learning. The students who were teachers themselves stated that they would recommend using team-based learning instead of the traditional lecture-based learning (Walker & Zheng, 2017).

Evaluation for Accountability

In team-based learning, team members are usually asked to evaluate the contributions of their fellow team members to ensure accountability within the teams (Yoon, Park, Myung, Moon, & Park, 2018). To assess the validity and reliability of the peer evaluation instrument in team-based learning courses at a medical school, Yoon et al. (2018) conducted a mixed methods research study at Seoul National University College of Medicine in South Korea. The study included 141 students who were divided into 18 groups in 11 team-based learning courses. Yoon et al. evaluated peer assessments from the participants. The peer evaluations identified the weakest and strongest group members and included the comments from the assessments. After each class that involved team-based learning, the students evaluated their teammates based on five criteria. The criteria were "(1) Did the peer prepare enough for the class? (2) Did the peer actively participate in the group discussion? (3) Did the peer contribute to the group activity? (4) Did the peer respect others in the group? (5) Did the peer show sincerity during the class?" (Yoon et al., 2018, p. 24). The students rated their peers beginning with an average of 10 points for each person. The students were required to rate at least one team member above 11 points, and at least one team member below nine points. The students were asked to provide a reason for giving the highest points and the lowest points for the students who were evaluated. The written comments were evaluated to make sure that the students completing the evaluation followed the correct instructions. Two researchers independently reviewed all of the comments and grouped the comments.

Researchers then compared their results and developed a final set of grouped comments. The reliability of the evaluation process was determined by using an intraclass correlation coefficient analysis, then the intraclass correlation coefficient analysis was compared between the different groups and examined to determine if the intraclass correlation coefficient analysis varied by the number of team members in the groups. In addition, the intraclass correlation coefficient analysis was examined to determine if the value changed during the time that the students were taking the course. Analysis of variance was used to analyze differences among the intraclass correlation coefficient analyses. The research determined that having six to seven students in each team was the best group size to encourage the students' interaction in the group while not compromising the reliability of the evaluations. Providing guidelines for the peer evaluation improved the validity and reliability of the evaluation tool. The study was completed using a single evaluation method at one school. Therefore, the results of this study may not be generalizable to other peer evaluation methods in other schools. The students were placed into the teams based only on their student number. If the groups were set up based on characteristics, such as the students' age, gender, or academic achievements, there may have been different results. Future research is needed to determine if any of the dynamics within the group may influence the findings of this research. Furthermore, future research could be conducted to improve the validity and reliability of the current peer evaluation methods (Yoon et al., 2018).

The purpose of the research conducted by Stein, Colyer, and Manning (2016) was to analyze the comments that students gave to their teammates to determine if the students held each other accountable for their parts in the team. In their mixed methods study, 211 students who had completed three courses at West University were divided into 35 teams. Peer assessments from the participants, which identified the weakest and strongest group members and the comments from the assessments, were evaluated. The assessments were part of the assignments for the team-based learning course that the students had taken. The average of the students' assessment scores were used to rank the students within their teams. Comments for the top-ranking team member and for the lowest ranking team member were selected and coded. Themes were determined based on the codes for the strongest and the weakest team members. The top ranked team members were listed as having a good work ethic, having initiative, being reliable, and being intelligent. Not missing class was the most common comment given to the top ranked team member. In contrast, the weakest team member was criticized for not coming to class and therefore could not participate in the teamwork. The weakest team member was also noted to be a complainer and to not be prepared. Almost all of the weak students were criticized for constantly being on their phone instead of participating in the class. Other comments that should be considered, but did not necessarily deal with rankings, was the way that the room was set up for group work and students who were quiet and shy. The findings indicated that the students gave their peers accurate evaluations. The study did not account for how peer feedback might influence team members' behaviors. The classroom set up may have had a negative impact on how the team members were able to communicate and work together. Future research would be needed to evaluate how team member assessments affect each of the team member's behavior during the semester. Research should be conducted to determine what resources could be used to help students who struggle with communication or who are shy succeed as a member of the team (Stein et al., 2016).

Teamwork and Engagement

Being able to work with others on a team is one of the top five characteristics for employers (Espey, 2018). In order to work well in a team, the team members must be able to communicate well, have frequent interactions with each other, and have mutual trust and respect for each other. When creating teams, understanding what characteristics that the team members should have to make a better team is important. Using teams in the classroom and teaching team skills can improve the motivation of the students, improve their communication skills, help them to learn the material, and improve their critical thinking skills. In Espey's (2018) research, she sought to determine if the make-up of the team members' gender, class level, and combination of students from within the state and from out of state contributed to the team performance and to the individuals' performance. She also wanted to determine if students learned more when they were actively participating and engaging in the course in a team setting and if the level of their contributions affected the amount of material that they learned. She conducted a qualitative study of 114 teams of 684 students who were in 17 sections of introduction to microeconomic theory courses that were taught from 2007 to 2016 at a four-year public university. The students' gender, class ranking, selected major, hometown, the students' grade point average, and peer evaluations for the end of the semester were collected and summarized. The team size and class size were collected. The maximum, lowest, and average grade point averages were calculated, and team grades were evaluated to determine if individual members' grade point average and team grades influenced the teams' performance. The peer evaluations were used to determine how well the teams worked together. Faculty members need to be aware of any characteristics of students within the teams that may contribute to the teams' success or failure. If problems between team members arise, they need to be handled as quickly as possible. Faculty also need to encourage all of the students in the teams to contribute to the teamwork. The faculty should explain to the students the importance of working together in teams. If teams are set up correctly, the teams' performance and the individuals' performance will improve. Evidence indicated that a greater diversity of the gender within a team helped individuals within the team to be more successful. If students can learn to work together in teams, their productivity in the course will improve, and they might have an advantage when looking for a job (Espey, 2018).

Research has shown that students who are actively engaged with the courses content are more successful in the course; therefore, encouraging students to be engaged in the course is an important part of teaching the course (Alvarez-Bell, Wirtz, & Bian, 2017). The use of team-based learning had been explored in many degree programs but had not been researched specifically for general chemistry courses. To determine undergraduate general chemistry students' feelings about student engagement, course instruction, and learning, Alvarez-Bell et al. (2017) conducted a quantitative research study with 111 students in a general chemistry II class at a four-year public university in eastern North Carolina. Using an online survey to collect data, 108 students responded to the teamwork and feelings sections of the survey, and 102 students responded to the class status section. The students volunteered to participate and were not compensated financially; however, they received extra credit in the course for participating. The survey included 81 items covering the students' demographics, engagement, instructional practices, and their perceived learning. The participants responded to the survey items using a five-point Likert-type scale. Data were analyzed using SPSS. An exploratory factor analysis was conducted on the items in the areas of engagement and instructional practices. Multiple linear regression was conducted with the demographics information. Bivariate correlation analyses were used for the students' self-reported amount of perceived learning. The students' feelings about the environment where they were learning and the amount of guidance that they received from their instructors were the best indicators of how well they would do. Learning was improved when the instructors acted as facilitators of learning instead of just being experts that gave out knowledge. When the students were encouraged to take active roles in their learning, they were more accountable for their own learning and the learning of their peers (Alvarez-Bell et al., 2017).

While evidence that team-based learning is effective for promoting learning in many areas, there has been only limited research that team-based learning supports other objectives, such as teamwork and communication skills. Wu et al. (2018) sought to evaluate the perceptions of medical educators on how well team-based learning met these objectives. They reviewed previously published studies and were able to identify 21 claims from other researchers of how well team-based learning met these objectives. These claims were divided into four categories, which included learning, behaviors, skills, and well-being. They distributed a quantitative questionnaire using a five-point Likert-type scale to medical educators who had taught using team-based learning and who were active members of the team-based learning collaborative listserv group. They received 50 responses. The responses for each of the claims was averaged and compared to the other claims. The participants agreed with the claims that team-based learning improved the students' behaviors and skills, promoted self-directed and active learning, peer-to-peer learning, and teaching. They also agreed that team-based learning encouraged teamwork, collaboration, communications among peers, and problem-solving skills (Wu et al., 2018).

Huang and Lin (2017) developed a flipped classroom module using team-based learning to promote active learning and participation in human resource management courses. Their mixed methods study included 104 students and two professors from two universities in Taiwan. Six students and two instructors were interviewed, and 102 students completed questionnaires. The questionnaire contained 18 questions and four measures, which included the team members' perception of the other team members' contributions, their perceived motivation, their perceived enjoyment, and their perceived learning. Descriptive statistics and inter-correlations were conducted on the quantitative data, and a regression analysis was conducted to determine the relationship between the variables. The researchers concluded that the effectiveness of team-based learning could affect the students' motivation to learn, their enjoyment in the course, and the outcomes of the flipped classrooms. Students learned better when they believed that their team members were dedicated to completing team projects. Two critical elements of the students' learning experience and the quality of their learning were the students' motivation when taking the course and their enjoyment when taking a flipped classroom team-based course. The researchers noted that, although the students cared about their teammates' contributions, their own motivation and enjoyment were what determined the learning outcomes. When the students were motivated to learn, they liked the flipped class team-based learning strategy and learned more course material. The students had more confidence in their ability to learn using the flipped team-based format. The conclusion was that, when the students enjoyed the team activities, they learned more course content (Huang & Lin, 2017).

The increased use of technology has included the use of audience response systems, or clickers, in the classroom. Johnson (2017) conducted a study to compare student learning in the team-based learning classroom with different team reporting methods. There had not been a study conducted to measure the effectiveness of these devices to learn the material. To compare the differences in students' learning when the students used the clickers verses when the students used small erase boards to answer their team questions in team-based learning, Johnson (2017) conducted a quantitative study that included 119 students who were enrolled in two biology courses at a small 4-year private liberal arts university in the southeastern United States. The students in the study were given pre- and post-exercise quizzes at the beginning and the end of the study. The amount of material that was learned by the students was measured by the difference between the pre- and post-scores. A paired samples *t*-test was conducted to compare the difference between traditional and technological reporting for each activity. Johnson found no statistical significant difference between the use of audience response systems and the white boards for each class. However, when the data for the classes were combined and the outliers were removed, a low statistical significance was found. A limitation of the study was that the study was conducted at a small four-year private university. Learning gains in the study may have been more evident in a team-based learning course with higher enrollment. Because this study was conducted during a single term, the results could not show the long-term success and cannot be generalized to other assessments. Future research should include a larger group of students with more teams. The pre- and post-quizzes should have more in depth questions. Future research could measure improvement in student knowledge and the students' feelings concerning using different reporting methods by the team (Johnson, 2017).

Effectiveness of Team-based Learning

The traditional instructional strategy of teaching engineering students has been lecture-based. Limited research exists on the effectiveness of using team-based learning for first-year engineering courses. Najdanovic-Visak's (2017) quantitative study included 115 students divided into 18 groups (i.e., 11 groups with six students and seven groups with seven students). The students' individual readiness assurance test, team readiness assurance test, application exercise scores, and a survey using a Likert-type scale with two open-ended questions was given twice during the semester. The individual readiness assurance test, team readiness assurance test, and exercise scores were averaged, and the survey totals were calculated. Statements from the survey were used to validate the scale value given by the students for the survey questions. The average team scores were higher than the average individual scores by 12%, and 67% of the teams had higher scores than the scores of their best team member. After the first two weeks, 50% of the students felt that they did not learn better using team-based learning. Five weeks after the course began, only 38% of the students felt this way, and, by the last evaluation, only 18% of students felt this way. This change may have been due to the students learning to work in groups as the course continued. Students reported that they liked working together in teams, getting to know their classmates, and learning from each other. The students did not like the pre-class reading and preparation. The students preferred the instructor to cover the material first. Some students also commented that they did not like working in groups. The study implied that possible benefits of using team-based learning in a first-year engineering course could be found. However, limitations of the study included that the findings may not be generalizable to larger universities or to complete courses because it was completed for a single module of a course at a small university (Najdanovic-Visak, 2017).

Pardamean et al. (2017) conducted a comparative quantitative research study to determine if the individual readiness assurance test used in team-based learning had an impact on the final exam scores of the course. The study included 96 students in two sections of an international business strategy course at the Swinburne University in Hawthorne, Australia, and 66 students in two sections of the basic accounting course at Bina Nusantara University in Jakarta, Indonesia. The individual readiness assurance test and the final exam scores in each of the courses were used for the study. The individual readiness assurance test was administered three times during the semester for the accounting course. The individual readiness assurance test was administered five times during the semester for the international business course. The test was administered after the students studied the material before the instructor covered the material with the students in the course. The test was administered during the first 15 minutes of the course. Following the individual readiness assurance test, the students completed the same test, a group readiness assurance test, within their predetermined groups in the course. The scores of the individual and group readiness assurance tests and the scores for the final exams were collected, and the means were calculated. The means of the individual readiness assurance tests were compared to the means of the final exams for each course. In both courses, the group readiness assessment test scores were always higher than the individual readiness assurance test scores, which confirmed that the students performed better when they worked in teams. The means of the individual readiness assurance test for both courses were nearly the same. The score for the international business strategy course was 60.62, and the score for the basic accounting course was 59.50. The results indicated that the students taking the international business strategy course made an improvement in their scores from taking the individual readiness assurance test and the final exam. An increase of 9% from the mean individual readiness assurance test score of 60.62 to the mean final exam score of 66.12 was reported in the international business strategy courses, but the scores in the accounting course did not change. The mean score for the individual readiness assurance tests and the final exams was 60. One of the reasons that the scores did not change could be that the accounting course only used team-based learning and the individual and group readiness assurance tests during the first half of the semester (Pardamean et al., 2017).

To determine the effect that team-based learning has on long-term retention of knowledge compared to retention of knowledge in a traditional curriculum, Emke, Butler, and Larsen (2016) conducted a quasi-experimental research using medical students at the Washington University School of Medicine. Medical students need to gain a tremendous amount of knowledge during the pre-clinical duration of their undergraduate medical education. Because this knowledge is the foundation for the students' clinical training, retention of the knowledge from their pre-clinical courses is critical. The students were divided into two groups of 122 students, which included a control group and an experimental group. Each group was divided into 18 teams with each team having six to seven students. The students completed a multiple-choice test covering the knowledge that they had learned four different times over 2 years. The students' performance on their test was compared to their prior tests to determine if any changes in their knowledge retention had occurred over time. The control group and the experimental group took the tests. A multivariate analysis of variance (MANCOVA) was conducted with the data. The group was the independent variable, the first knowledge test was the covariate, and the three other tests that were given over the 2 years were the dependent measures. A second MANOVA was conducted using the group as the independent variable. The changes in the scores represented the gain from the first knowledge test to the three following tests, which served as the dependent measures. Following the MANOVAs, the difference in the performance between the two groups for each test was compared using independent *t*-tests. All three analysis produced the same results. Students who did not complete all of the evaluation tests and students who were PhD students were excluded from the final results. After these exclusions, 40 students remained in the control group, and 46 students remained the intervention group. Many benefits existed when using team-based learning; however, the results did not indicate that long-term retention of knowledge was better than using a traditional lecture format. Although there was better retention of knowledge for the short term, the researchers concluded that, without continued practice, the knowledge that the students gained was easily lost (Emke et al., 2016).

Summary

Remington et al. (2017) and Kirkpatrick (2017) found that, although students liked the application exercises and team discussions used in team-based learning, the students did not like the work that was required to prepare for the course. Both researchers found that the students perceived that they were more engaged in the class and learned more from the discussions with their peers. The students also perceived that using team-based learning gave them more confidence in their ability to make decisions. Frame et al. (2015) found that the earlier the students were introduced to team-based learning, the more they liked it. The students also developed a more positive view of teams and teamwork by the end of the semester than students who were taught using traditional lecture-based learning. Jeno et al. (2017) concluded that using team-based learning increased the students' intrinsic motivation, external regulation, and competence using small discussion groups, immediate feedback from the readiness assurance tests, and the instructors' increased guidance and support. Burgess et al. (2017) also reported that the students preferred the smaller size of the groups, the readiness assurance tests, the immediate feedback, and the time efficiency of the team-based learning approach.

Lasserre and Szostak (2011) found that the students who took the course using team-based learning increased their score from 54% to 75.5% and the withdrawal rate went down from more than 30% to 6.4%. Comeford (2016) also found an improvement in the failure rate after implementing team-based learning from 31% to 19%. However, Emke et al. (2016) found that, although there was better retention of knowledge in the short term when using team-based learning, the long-term retention of knowledge was similar for team-based learning and lecture-based learning. Figure 2 presents a concept analysis chart for studies related to team-based learning.

Study	Purpose	Participants	Design/ Analysis	Outcomes
Burgess et al. (2017)	To compare students' perceptions of taking a course using team-based learning verses taking a course using problem-based learning.	169 first-year medical students	Mixed methods: Quantitative data analyzed using descriptive statistics and qualitative data analyzed using thematic analysis	• Team-based learning benefits included smaller groups, readiness assurance tests, immediate feedback, and time efficiency. • Problem-based learning problems included inconsistent tutors in their level of expertise, unclear directions, and large group size.
Comeford (2016)	To use team-based learning to improve the retention rate in introduction to chemistry classes.	Five sections of intro to chemistry prior to team-based learning (*N* = 192) and nine sections after implementing team-based learning (*N* = 285)	Quantitative: data analyzed using Fisher's exact test	• The failure rate after using team-based learning decreased from 31% to 19%.

Study	Purpose	Participants	Design/ Analysis	Outcomes
Emke et al. (2016)	To determine the effect that team-based learning had on long-term retention of knowledge compared to learning in a traditional curriculum.	244 medical students	Quasi experimental: MANCOVA and independent *t*-tests	• Short-term retention of knowledge was better using team-based learning. • Long-term retention of knowledge was similar to traditional lecture-based format.
Frame (2015)	To evaluate student perceptions of team-based learning compared to traditional lecture-based learning.	111 first-year pharmacy students	Quantitative: SPSS chi-square tests, Wilcoxon signed rank test, Mann-Whitney U test	• When the students took the team-based learning course influenced the students' view of how useful the teaching strategy was. • Team assignments were important due to an increase in team-based, patient-centered health care.
Jeno et al. (2017)	To determine the effects of team-based learning on motivation and learning.	24 biology students	Quasi-experimental, pretest/posttest using *t*-tests and path analysis	• Team-based learning increased students' intrinsic motivation, external regulation, and competence.
Kirkpatrick (2017)	To determine the students' feelings about using team-based learning in a computer science course.	111 introduction to programming students: 55 team-based learning and 56 traditional	Explanatory mixed-methods observational study	• Team-based learning was a good use of class time. • Class discussions helped students learn the material better.

Study	Purpose	Participants	Design/ Analysis	Outcomes
		lecture-based learning.		
Lasserre & Szostak (2011)	To determine the effectiveness of using team-based learning to improve the withdrawal rate in introduction to programming courses.	294 students over 4 years of introduction to computer programming courses at UBC Okanagan	Quantitative survey	• The number of students scoring 50% or higher on the final exam increased from 54% to 75.5%. • The withdrawal rate in the courses decreased from over 30% to 6.4%.
Remington et al. (2017)	To compare students' attitudes and perceptions about team-based learning to lecture-based learning.	30 students in a pharma-cotherapeutic s course	Qualitative using a grounded theory coding method	• Students were able to retain the material better using team-based learning. • The students were more engaged in the course and learned more content from the discussions with their peers. • Using team-based learning improved the test scores of the students when compared to lecture-based learning courses.

Figure 2. Concept Analysis Chart for studies related to team-based learning

Conclusion

The introduction to programming course is considered a challenging and difficult course for many beginning computer science students (Bennedsen & Caspersen, 2019;

Watson, 2014). As a result, high failure and withdrawal rates from these courses have been reported. The primary issue that students have when learning to program is the multiple skills that have to be learned at the same time. Specifically, students have to learn the syntax and rules of the programming language, and they have to develop problem-solving skills. Traditional lecture and lab-based strategies that are used to teach these introduction to programming courses concentrate more on teaching the syntax of the language instead of the problem-solving skills that are needed to complete programming problems. A different instructional strategy that incorporates problem-solving skills along with syntax and rules could improve the success rates in the course.

The traditional strategy for teaching an introduction to programming course is giving lectures that cover the theory and concepts and then assigning lab assignments for the students to complete at home so that the students can put the concepts into practice (Hegazi & Alhawarat, 2015). The research indicated that there might be a better way to teach the introduction to programming course so that the students can understand and apply the concepts successfully. Team-based learning has been shown to improve the success rate in these courses and the amount of skills that the students are able to learn and retain (Nikooravesh, Parpoochi, & Davoudi, 2016). The purpose of this study was to determine if differences existed in the numerical unit test scores between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format.

CHAPTER III

METHODOLOGY

A problem of low success rates in the introduction to programming course currently exists. The low success rates affect the graduation rates in the STEM degree programs (Chen, 2015). The STEM degree programs are made up of the computer science, information technology, mathematics, and engineering majors. The problem especially affects the computer science and information technology degree programs because the introduction to programming course is considered the gateway course for these two majors (Chen, 2015). That problem, specifically, is the low success rate in the introduction to programming course at a small southwest Georgia university. This chapter will include descriptions of the research design and the methodology (i.e., the participants, instrumentation, intervention, data collection, and data analysis) that were used in the study.

The purpose of the study was to determine if differences existed in the unit test scores between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format. To address this purpose, a causal-comparative research study was conducted. The research questions for this study were:

RQ1. What is the difference in students' knowledge of conditional expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university?

H_{01}: There is no difference in students' knowledge of conditional expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university to a statistically significant degree.

H_{a1}: There is a difference in students' knowledge of conditional expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university to a statistically significant degree.

RQ2. What is the difference in students' knowledge of looping expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university?

H_{02}: There is no difference in students' knowledge of looping expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to

programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university to a statistically significant degree.

H_{a2}: There is a difference in students' knowledge of looping expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university to a statistically significant degree.

Research Design

Quantitative research involves data in the form of numbers that can be analyzed later. Quantitative studies can be experimental or non-experimental. Experimental studies are based on treatments that are given to the research participants to observe the effects of the treatments on the experimental group. An experimental design could not be used because the sample for the current study could not be randomly assigned. Non-experimental studies, such as the current study, are known as descriptive studies because the researcher is describing previous data that has been collected and analyzed (Patten & Newhart, 2018). A correlational study could not be used because correlation studies cannot determine the cause and effect relationship between the variables.

A qualitative research design could not be used to answer the research questions for this study. Qualitative research methods are used to answer questions about the personal experience of the participants of the study and are not used to compare numbers, such as counting or measuring ranges (Hammarberg, Kirkman, & de Lacey, 2016).

A causal-comparative quantitative research design was used for the study because the groups were established and the intervention was implemented prior to the research being conducted (Salkind, 2010). A causal-comparative research design was used because the variables were studied in retrospect. The causal-comparative research design sought to determine if there was a difference between existing conditions between different groups. The individuals in the study were not randomly assigned to one of the two groups because the participants' enrollment in the courses was pre-existing based on the students' registration into the course. The participants for this study were intact groups, meaning the groups were established based on the students' choice of major (Lomax & Hahs-Vaughn, 2012). A disadvantage of the causal-comparative research design is the limited control that the researcher has over the study because the groups were formed prior to the beginning of the study.

The study examined the differences between the dependent variables between each independent variable. The independent variable was defined as the group (i.e., the control group included the students who were taught using a traditional lecture and lab class and the treatment group included the students who were taught using team-based learning). The dependent variable was the numerical unit test scores from the course. The unit tests covered the topics of conditional expressions (i.e., Unit 3) and looping expressions (i.e., Unit 4).

Role of the Researcher

The researcher for the study was both an observer and the instructor in the study. There was only one instructor at the university who taught the course. The researcher had taught the course each semester for over 20 years. The researcher held a Bachelor of

Science in Information Technology and a Master of Science in Computer Science. The researcher also had 10 years of experience working in the computer science and information technology fields prior to becoming an educator. No participant in the study had a direct relationship with the researcher other than a student-teacher relationship. To avoid a conflict of interest for this study, numerical unit test scores were pulled after final grades had been posted for the semester.

Participants

The population of the study included all of the students who were required to take the introduction to programming course at the selected institution. Students who majored in computer science, information technology, mathematics, and engineering were required to take the course. Table 1 displays the number of students included in each of the majors by semester.

Table 1

Frequency and Percentage of Undergraduate Degree Majors by Group

	Fall 2018		Spring 2019		Fall 2019	
Major	*n*	%	*n*	%	*n*	%
Computer Science	72	39.1	59	39.1	62	31.2
Information Technology	56	30.4	49	32.5	97	48.7
Mathematics	14	7.6	10	6.6	14	7.0
Engineering	42	22.8	33	21.9	26	13.1
Total	184	100	151	100	199	100

The sample for the study included all of the students who took the introduction to programming course during the Fall 2018, Spring 2019, and Fall 2019 semesters and who completed all four of the unit tests. Each of the participants were required to take the course as a part of their major's program of study. The majority of the participants were classified as freshmen. The data were de-identified prior to data analysis, so the specific number of freshman students was unknown. Due to the high demand of the course, only students required to take the course were allowed to register for the course, which included students who majored in information technology, computer science, mathematics, or engineering. The control group consisted of the students who took the introduction to programming course using a traditional lecture and lab-based teaching strategy in Fall 2018 and Spring 2019. The treatment group consisted of the students who took the introduction to programming course using a team-based teaching strategy in Fall 2019.

Instrumentation

The instructor created the unit tests. Each unit test covered the most important concepts taught in that unit. The questions for the unit tests were created using the information in the textbook, the questions at the end of the chapters, and the examples from the book that were covered in the class. The textbook that was used for the class was *C# Programming Fifth Edition, From Problem Analysis to Program Design* (Doyle, 2015). The unit tests were given during an in-class session. The students were given the entire class period of 75 minutes to complete the test.

The unit test for programming terminology and data types (i.e., Unit 1) aligned with the Chapter 1 and Chapter 2 content from the course textbook. The test consisted of 15 short-answer questions, and five questions that required the students to write the code to declare variables, display output, and increment a variable. Each question was worth five points. Partial credit of two points was given for short-answer questions where at least half of the answer was correct and the answer showed that the student understood the concept but did not explain the answer correctly. For the unit test covering programming terminology and data types, the students could earn a maximum of 100 points.

The unit test for methods and behaviors (i.e., Unit 2) aligned with the Chapter 3 content from the course textbook. The test consisted of one true/false question, 10 short-answer questions, and nine questions that required the students to write the code to call a method or to write the code for the method. Each question was worth five points. Partial credit of two points was given for short-answer questions where at least half of the answer was correct and the answer showed that the student understood the concept but did not explain the answer correctly. For the unit test covering methods and behaviors, the students could earn a maximum of 100 points.

The unit test for conditional expressions (i.e., Unit 3) aligned with the Chapter 5 content from the course textbook. The test consisted of 18 short-answer questions, two questions that required the students to write a conditional expression, and a bonus question that required the students to write the code for a switch statement. Each test question was worth five points, and the bonus question was worth 10 points. Partial credit of two points was given for short-answer questions where at least half of the answer was correct and the answer showed that the student understood the concept but did not explain the answer correctly. For the unit test covering conditional expressions, the students could earn a maximum of 110 points.

The unit test for the looping expressions (i.e., Unit 4) aligned with the Chapter 6 content from the course textbook. The test consisted of four true/false questions, five multiple-choice questions, 14 short-answer questions, and two questions that required the students to write the code to perform a loop. Each question was worth four points. Partial credit of two points was given for short-answer questions where at least half of the answer was correct and the answer showed that the student understood the concept but did not explain the answer correctly. For the unit test covering looping expressions, the students could earn a maximum of 100 points.

Intervention

Team-based learning reported an increase in the completion and pass rate of introduction to programming courses by 20% as compared to the same course being taught in a traditional lecture-based format (Lasserre & Szostak, 2011). In order to implement team-based learning, the instructor formed teams of four students at the beginning of the semester. The teams remained together for the duration of the course. Having the teams stay together allowed the teams to learn to work together and learn from each other. The teams were assigned randomly by the learning management system initially. The instructor then examined the assigned teams to ensure that they included students with different experiences and strengths. To provide accountability within the teams, a peer assessment accounted for 50% of the students' assignment grade. The students' assignment grade accounted for 30% of the students' overall final grade in the course. The peer assessment was made up of three parts, including the Assessment of Individual Team Member's Team Working Skills (MacVarish & Cox, 2015; see Appendix A), the Assessment of Team Effectiveness (MacVarish & Cox, 2015; see Appendix B), and the Group Project Evaluation (LaMorte, 2015; see Appendix C). Permission to use the Group Project Evaluation was requested from Dr. Wayne LaMorte and was granted on June 7, 2020, via email (see Appendix D). Permission to use the

Assessment of Individual Team Member's Team Working Skills and the Assessment of Team Effectiveness was requested from Dr. Kathleen MacVarish and Dr. Harold Cox and was granted on June 8, 2020, via email (see Appendix E).

At the beginning of each unit, students completed a reading assignment and were given practice questions and the answer key to review. During the in-class session, a review of the material that was covered in the unit occurred, and students were encouraged to ask questions over the material. At the beginning of the following in-class session, students took an individual readiness assurance test that contained 20 multiple-choice questions covering the key concepts of the unit. Each question had four possible answers. The test was administered using the online course management system used by the university. The students were given 15 minutes to complete the test. The test was graded automatically by the course management system as soon as it was submitted, and the grade was displayed for the student. After all of the students had completed the individual readiness test, the team readiness assurance test was completed with their assigned team members. The team readiness assurance test contained the same questions that were presented in the same order as the individual readiness assurance test. The teams were given 30 minutes to complete the team readiness assurance test and were encouraged to discuss the questions and answers with each other to ensure that each member understood the questions and the answers. The scores from the individual test and the team test were averaged together for each student. Following the readiness assurance tests, the teams were given the same programming assignment. Each team was given the same amount of time to complete and submit the assignment. Following the assignment submission, the students took a unit test individually. For this study, the scores for the unit tests were compared to determine if differences existed in the numerical unit test scores between the students who took the course using a traditional lecture and lab format and the students who took the course using a team-based learning format. Figure 3 depicts the sequence of the team-based learning teaching strategy.

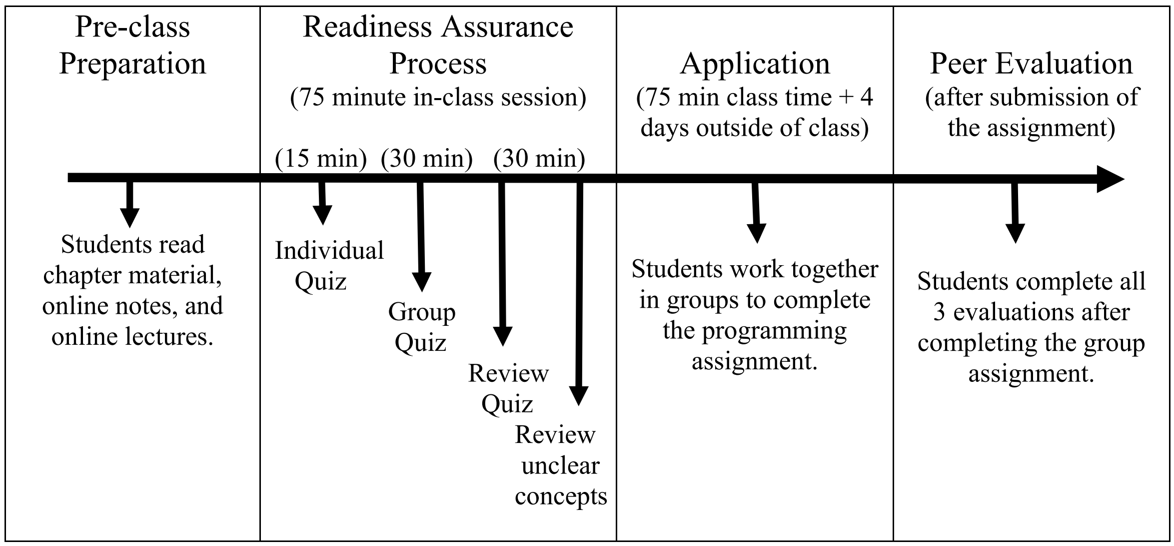

Figure 3. Team-based learning teaching strategy process.

Data Collection

For each semester, the first half of the course was taught using a traditional lecture and lab-based teaching strategy. The first half of the semester covered the introduction and basic concepts that the students should know in order to begin to write computer programs, such as syntax, variable types, and methods (i.e., Unit 1 and Unit 2). For the control group, the second half of the semester was taught using a traditional lecture and lab-based teaching strategy. For the treatment group, the second half of the semester was taught using a team-based learning strategy. The topics for the second half of the semester were conditional expressions (i.e., Unit 3) and looping expressions (i.e., Unit 4). The participants for each group completed an assignment and a test over each unit. The numerical scores from Unit 1 and Unit 2 tests from the first half of the semester were included in the study as covariates. The covariates were included to determine if the students' foundational knowledge impacted the students' Unit 3 and Unit 4 numerical test scores. The unit tests for each semester were the same and, therefore, could be used for comparison. Because the assignments differed for each semester, they could not be included in the study.

At the end of each unit, the students took a unit test consisting of 20 to 25 questions, which included true/false questions, short-answer questions, and questions that required the students to write code based on the content covered in the unit. Each student took the same test, and the instructor graded each test using the same answer key. During the following class session, the students and the instructor reviewed the test. After the students and the instructor reviewed the test, the test scores were recorded in the learning management system that was used for the course, and the scores were recorded in an Excel spreadsheet.

Numerical unit test scores from Fall 2019, which utilized team-based learning, were collected and compared to the same unit tests from Fall 2018 and Spring 2019, which utilized a traditional lecture and lab-based teaching strategy. The standard practice of the instructor/researcher was to retain the course grades in the learning management system and in an Excel spreadsheet. The unit tests were stored in the instructor's office in a locked cabinet for four semesters. After the fourth semester, the tests were shredded and disposed in a secure container. The individual test scores were stored for the long-term as part of the course instructor's records. Because no identifying data were collected from the participants and the instructor collected the data as part of normal classroom activities, the researcher requested a waiver for the informed consent from the Columbus State University Institutional Review Board (see Appendix F).

Data Analysis

Three new Excel spreadsheets were created to compile and code the students' numerical unit test scores from the three original Excel spreadsheets that were created for each semester included in the study. Any identifying information, including the students' identification number and name, were removed before entering the data in SPSS. The students were identified using a sequential numbering system.

In order to categorize the instructional strategies, the unit tests, and the semesters, dummy coding was used. Dummy coding provided a way to convert nominal values used to describe the specifics of the data into ordinal values that could be used in statistical analysis (Lomax & Hahs-Vaughn, 2012). Each of the three datasets were dummy coded. The independent variable was dummy coded using a 1 for the traditional lecture and lab-based class format and a 2 for the team-based learning format. The covariate variables were dummy coded to represent the content of the unit test. The unit test for programming terminology and data types (i.e., Unit 1) was coded using a 1, and the unit test for methods and behaviors (i.e., Unit 2) was coded using a 2. The dependent variable was dummy coded to represent the content of the unit test. The unit test for conditional statements (i.e., Unit 3) was coded using a 3, and the unit test for looping expressions (i.e., Unit 4) was coded using a 4. The semester was dummy coded. Fall 2018 was coded as a 1, Spring 2019 was coded as a 2, and Fall 2019 was coded as a 3. The three Excel spreadsheets, which included the unit test scores for Fall 2018, Spring

2019, and Fall 2019, were merged into a single Excel spreadsheet so that the data could be entered in SPSS.

Levene's Test of Equality of Variances was conducted to test the equality of variances between the unit test scores for each group. Equality of variances needed to be assessed to determine if the differences between the data groups were due to chance, or if the variances were actually different (Filliben & Heckert, n.d.; Lomax & Hahs-Vaughn, 2012).

Descriptive statistics were conducted for each unit test by group. The descriptive statistics included the mean score for each unit test, as well as the standard deviation and the range for each unit test score. The ANCOVA controlled for the effects of other variables that were not of primary interest, known as covariates. The numerical unit test scores for Unit 1 and Unit 2 served as the covariates for this study.

A series of one-way ANCOVAs was performed because each ANCOVA tested a different dependent variable. An ANCOVA compares the means of two or more groups and tests the hypothesis made about the differences between the means of the groups. Each one-way ANCOVA compared the means of each unit test by group (i.e., teaching strategy). A one-way ANCOVA was used because there was only one independent variable with two levels and one dependent variable. A disadvantage of using an ANCOVA was if the null hypothesis is rejected, the result proves that at least one group differs from the others (Johnson & Christensen, 2017).

To answer Research Question 1, the independent variable was group (i.e., the control group who completed the course with a traditional lecture and lab-based learning format or the treatment group who completed the course using a team-based learning format), and the dependent variable was the numerical unit test scores covering conditional expressions (i.e., Unit 3). The numerical test scores from Unit 1 and Unit 2, which were administered during the first half of the semester, served as covariates. To answer Research Question 2, the independent variable was group, and the dependent variable was the numerical unit test scores covering looping expressions (i.e., Unit 4). The numerical test scores from Unit 1 and Unit 2, which were administered during the first half of the semester, served as covariates. The findings for each research question will be presented in table format in Chapter IV.

Summary

The goal of this chapter was to outline the research method used to answer the research questions. A discussion of the study participants, procedures, data collection, and data analysis outlined the specifics of how the study was conducted. A causal-comparative quantitative research design was conducted to answer the research questions. All study participants contributed to the study through the inclusion of their numerical unit test scores. The numerical scores from each unit that was taught during Fall 2019, which utilized team-based learning, were collected and compared to the same unit tests from Fall 2018 and Spring 2019, which were taught using a traditional lecture and lab-based teaching strategy. The numerical scores from Unit 1 and Unit 2 tests from the first half of the semester served as covariates. A series of one-way ANCOVAs was conducted to determine if there were differences between the means of the unit tests (i.e., Unit 3 and Unit 4) between the groups.

CHAPTER IV

RESULTS

A problem exists in the STEM degree programs, particularly in computer science and information technology, because there is a low success rate in the introduction to programming course, which is the gateway course for four STEM degree programs (i.e., computer science, information technology, mathematics, and engineering). That problem, specifically, is the low success rate in the introduction to programming course at a small southwest Georgia university. This problem affects the growing demand of computer science graduates because the low success rates in the introduction to programming courses leads to a low retention rate in the computer science degree majors (Chen, 2015). In Beaubouef and Mason's (2005) research, team-based learning was shown to improve the success rate in the introduction to programming course. Also, because team-based learning requires the teams within the course to design, write, and test computer programs together, the students have opportunities to practice writing code, managing their time, and sharing their skills (Wu et al., 2018).

Purpose of the Study

The purpose of this study was to determine if differences existed in the numerical unit test scores between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format. An improvement in the unit test could improve success rates of the students in the introduction to programming course and improve the retention rates of students in the computer science major. The independent variable was defined as the group (i.e., the control group who was taught using a traditional lecture and lab-based class and the treatment group who was taught using team-based learning). The dependent variable was defined as the numerical unit test scores in the course. The unit tests covered the topics of conditional expressions (i.e., Unit 3) and looping expressions (i.e., Unit 4). The covariates were the unit tests that covered general programming terminology and data types (i.e., Unit 1) and methods and behaviors (i.e., Unit 2).

Participants

The population of the study included all of the students who were required to take the introduction to programming course. Students who majored in computer science, information technology, mathematics, and engineering were required to take the course. The sample for the study included the students who took the introduction to programming course during the Fall 2018, Spring 2019, and Fall 2019 semesters and who completed all four of the unit tests. The majority of the participants were considered freshman. The exact number was unknown given the data were de-identified. Each of the participants were required to take the course as a part of their major's program of study. Due to the high demand of the course, only students required to take the course were allowed to register for the course, which included students who majored in information technology, computer science, mathematics, or engineering.

The control group consisted of the students who took the introduction to programming course during the Fall 2018 and the Spring 2019 semesters. The control group was taught using a traditional lecture and lab-based teaching strategy. The treatment group consisted of the students who took the introduction to programming course during the Fall 2019 semester. The treatment group was taught using a team-based teaching strategy.

Pre-existing data were collected from the control group and the treatment group for the semesters included in the study. Only the students who completed all four of the unit tests were included in the study. In Fall 2018, 29 students enrolled in the course. One student did not complete the Unit 3 Test and the Unit 4 Test. Ten of those students withdrew from the course. Of the 10 students who withdrew from the course, two of the students did not complete any of the unit tests, two of the students completed the Unit 1 Test, and six of the students completed the Unit 1 Test and the Unit 2 Test, which resulted in 18 students completing the four unit tests. In Spring 2019, 31 students enrolled in the course. Three of the students only completed the Unit 1 and Unit 2 Tests, which resulted in 28 students completing the four unit tests. In Fall 2019, 26 students enrolled in the course. One student only completed the Unit 1 Test, and one student only completed the Unit 1 Test and the Unit 3 Test, which resulted in 24 students completing the four unit tests. The data were de-identified prior to data analysis. As shown in Table 2, unequal variance was found in the data for the Fall 2018 section; therefore, only the data from Spring 2019 and Fall 2019 could be included in the data analysis. The Spring 2019 students served as the control group, and the Fall 2019 students served as the treatment group.

Table 2

Descriptive Statistics for the Unit Tests by Section

Section	Test	*N*	*min*	*max*	*M*	*SD*
Fall 2018	Unit 1	18	10	98	58.11	24.65
	Unit 2	18	25	109	69.56	24.60
	Unit 3	18	13	105	75.61	25.39
	Unit 4	18	18	96	64.28	19.86
Spring 2019	Unit 1	28	16	98	66.11	23.97
	Unit 2	28	45	97	73.39	13.42
	Unit 3	28	47	110	86.82	14.60
	Unit 4	28	31	100	77.46	18.06
Fall 2019	Unit 1	24	32	97	70.71	19.76
	Unit 2	24	32	94	68.54	17.10
	Unit 3	24	52	106	88.86	12.74
	Unit 4	24	44	100	77.50	14.89

Findings

Covariates

The Unit 1 Test (i.e., programming terminology and data types) and the Unit 2 Test (i.e., methods and behaviors) served as covariates to determine if the students' foundational knowledge affected the students' performance on the Unit 3 Test (i.e., conditional expressions) and the Unit 4 Test (i.e., looping expressions) during the second half of the semester.

When Levene's Test of Equality of Error Variances (Levene, 1960) was conducted to determine if the assumption of homogeneity was met using the Unit 1 Test (i.e., programming terminology and data types) as the covariate and the Unit 3 Test (i.e., conditional expressions) as the dependent variable, the results indicated that the difference in variance was not statistically significant, $F(1, 50) = 3.35$; $p = .08$. Table 3 presents the descriptive statistics for the Unit 1 Test by group.

Table 3

Descriptive Statistics for Unit 1 Test Scores by Group

Group	*N*	*min*	*max*	*M*	*SD*
Control	28	16	98	66.11	23.97
Treatment	24	32	97	70.71	19.76

When Levene's Test of Equality of Error Variances (Levene, 1960) was conducted to determine if the assumption of homogeneity was met using the Unit 2 Test (i.e., methods and behaviors) as the covariate and the Unit 3 Test (i.e., conditional expressions) as the dependent variable, the results indicated that the difference in variance was statistically significant, $F(1, 50) = 6.07$; $p = .02$. The assumption of homogeneity was not met. Therefore, the Unit 2 Test scores could not be used as a covariate because the variance in Unit 2 Test was not equal between the groups. Table 4 presents the descriptive statistics for the Unit 2 Test by group.

Table 4

Descriptive Statistics for Unit 2 Test Scores by Group

Group	*N*	*min*	*max*	*M*	*SD*
Control	28	45	97	73.39	13.41
Treatment	24	35	94	68.54	17.10

Research Question 1

The first research question asked if a difference existed between the students' knowledge of conditional expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university. Descriptive statistics, including mean, standard deviation, and range, were conducted for the Unit 3 Test scores for each group. For the control group, the mean was 73.39 with a standard deviation of 13.42, and the scores ranged from 47 to 110. For the treatment group, the mean was 68.54 with a standard deviation of 17.10, and the scores ranged from 52 to 106. Unit 3 Test included a 10-point bonus question, which allowed the numerical scores to exceed 100.

Levene's Test of Equality of Variances (Levene, 1960) was conducted to determine if the Unit 3 Test scores met the assumption of homogeneity. The result indicated that the difference in variances was not statistically significant [$F(1, 50) = 1.20$; $p = .28$]. Therefore, the assumption of equal variances was met.

A one-way ANOVA was conducted to determine if a difference existed between the group of students who served as the control group and the group of students who served as the treatment group (i.e., team-based learning). The F test for between-subjects effects for the Unit 3 Test scores indicated that there was not a statistically significant difference between the groups because the p value exceeded the .05 criteria level [$F(1, 50) = 0.29$; $p = .59$].

After conducting the ANOVA, an ANCOVA was conducted to determine if a difference existed between the group of students who served as the control group and the group of students who served as the treatment group with the Unit 1 Test scores serving as the covariate and the Unit 3 Test scores serving as the dependent variable. Levene's Test of Equality of Error Variances (Levene, 1960) was conducted to determine if the Unit 3 Test scores met the assumption of homogeneity when the Unit 1 test score was included as the covariate. The result indicated that the difference in variances was not statistically significant [$F(1, 50) = 3.35$; $p = .07$]. Therefore, the assumption of equal variances was met.

A one-way ANCOVA for the Unit 3 Test scores with the Unit 1 Test scores as the covariate indicated that there was not a statistically significant difference between the groups because the *p* value exceeded the .05 criteria level [$F(1,49) = .01$; $p = .94$]. The means along with the estimated marginal means for the Unit 3 Test, which are slightly higher with the Unit 1 Test scores serving as the covariate, are shown in Table 5 by group.

Table 5

Means and Estimated Marginal Means for the Unit 3 Test Scores by Group

Group	Actual Mean	Estimated Marginal Mean
Control	86.82	87.66
Treatment	88.88	87.90

Research Question 2

The second research question asked if a difference existed in the students' knowledge of looping expressions as measured by a unit test between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format while controlling for the students' knowledge in programming terminology and data types in a small southwest Georgia university. Descriptive statistics, including mean, standard deviation, and range, were conducted for the Unit 4 Test scores for each group. For the control group, the mean was 77.46 with a standard deviation of 18.06, and the scores ranged from 31 to 100. For the treatment group, the mean was 77.50 with a standard deviation of 14.88, and the scores ranged from 44 to 100.

Levene's Test of Equality of Variances (Levene, 1960) was conducted to determine if the Unit 4 Test scores met the assumption of homogeneity. The result indicated that the difference in variances was not statistically significant [$F(1, 50) = 0.51$; $p = .48$]. Therefore, the assumption of equal variances was met.

A one-way ANOVA was conducted to determine if a difference existed between the group of students who served as the control group and the group of students who served as the treatment group (i.e., team-based learning). The *F* test for between-subjects effects for the Unit 4 Test scores indicated there was not a statistically significant difference between the group because the *p* value exceeded the .05 criteria level [$F(1, 50) = 0.00$; $p = .99$].

After conducting the ANOVA, an ANCOVA was conducted to determine if a difference existed between the group of students who served as the control group and the group of students who served as the treatment group with the Unit 1 Test serving as the covariate and the Unit 4 Test scores serving as the dependent variable.

Levene's Test of Equality of Error Variances (Levene, 1960) was conducted to determine if the Unit 4 Test scores met the assumption of homogeneity when the Unit 1 Test scores were included as the covariate. The result indicated that the difference in variances was not statistically significant [$F(1, 50) = 0.01$; $p = .92$]. Therefore, the assumption of equal variances was met.

A one-way ANCOVA for the Unit 4 Test scores with the Unit 1 Test scores as the covariate indicated that there was not a statistically significant difference between the group because the *p* value exceeded the .05 criteria level [$F(1,49) = 0.13$; $p = .72$]. The means along with the estimated marginal means for the Unit 4 Test scores, which were slightly lower with the Unit 1 Test scores serving as the covariate, are shown in Table 6 by group.

Table 6

Means and Estimated Marginal Means for Unit 4 Test Scores by Group

Group	Actual Mean	Estimated Marginal Mean
Control	77.46	78.19
Treatment	77.50	76.66

Summary

The purpose of the study was to determine if differences existed in the numerical unit test scores between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format. The sample for the study included the students who took the introduction to programming course during Spring 2019 (i.e., control group) and Fall 2019 (i.e., treatment group) and who completed all four of the unit tests. A series of one-way ANCOVAs was performed to determine if there were differences between the means of the unit tests used for the study (i.e., Unit 3 Test and Unit 4 Test) between the groups while controlling for the students' knowledge in programming and data types (i.e., Unit 1 Test). The analysis of the data indicated that there were not any statistical differences in the Unit 3 Test scores (i.e., conditional expressions) and in the Unit 4 Test scores (i.e., looping expressions) between the group of students who took the introduction to programing course using a traditional lecture and lab-based class format and the students who took introduction to programing using a team-based learning format. Chapter V will present the analysis of the findings.

CHAPTER V

DISCUSSION

Summary of the Study

A problem exists in the STEM degree programs, particularly in computer science and information technology, because there is a low success rate in the introduction to programming course, which is the gateway course for four STEM degree programs. That problem, specifically, is the low success rate in the introduction to programming course at a small southwest Georgia university. This problem affects the growing demand of computer science graduates because the low success rates in the introduction to programming courses leads to a low retention rate in the computer science degree majors (Chen, 2015). Hawi (2010) found that the learning strategy utilized in the introduction to programming courses was the main reason that students gave for the high failure rate in the course.

The purpose of this study was to determine if differences existed in the numerical unit test scores between students who took introduction to programming using a traditional lecture and lab-based class format and students who took introduction to programming using a team-based learning format. An improvement in the unit test scores could improve success rates of the students in the introduction to programming course and improve the retention rates of students in the computer science major.

The research design for the study was a causal-comparative research design, which determined if the independent variable (i.e., control group and treatment group) had an effect on the dependent variables, Unit 3 Test scores (i.e., conditional expressions) and Unit 4 Test scores (i.e., looping expressions). The Unit 1 Test scores over basic programming terminology and data types served as the covariate. The students' unit test scores were compared using an ANCOVA. The study sought to fill a gap in the literature examining teaching strategies to improve student success in the introduction to programming course. This study examined an approach to teaching the introduction to programming course using team-based learning, which may lead to improved success rates. However, there were not any statistically significant differences in the numerical unit test scores between the students who took the introduction to programming course using a traditional lecture and lab-based format and the students who took the introduction to programming course using a team-based learning format.

Analysis of the Findings

The introduction to programming course is typically the first course that students take in the computer science and information technology degree programs (Peterson et al., 2016). These students are adjusting from a pedagogy learning style to an andragogy learning style (Pappas, 2013). The changes in these learning styles, along with the students' need to learn new and sometimes more difficult material, can have an effect on the students' ability to be successful in the course (Celli & Young, 2017). In addition, when students fail the course, they tell other students, which spreads the thought that the course is very difficult. Therefore, when new students come into the class, they have the preconceived notion that the course is very difficult (Hegazi & Alhawart, 2015).

Andragogy (Knowles, 1984), the adult learning theory, was used as the theoretical framework for this study because there is no learning theory developed specifically for students between the ages of 18 and 25. Team-based learning was designed specifically to be used when teaching adult students. The average student age of the participants in the current study was 23.7 (College, n.d.b); therefore, the theory of andragogy along with team-based learning was used for the current study.

When the unit test scores for all three semesters were compared, the mean for Fall 2018 (M = 66.8) was 10 points lower than the means for Spring 2019 (M = 75.9) and Fall 2019 (M = 76.4). Due to unequal variance, the data from Fall 2018 were not included in the current study. At the selected institution, students are accepted into the computer science program each semester. Typically, the introduction to programming course is taken either the first or the second semester of the students' enrollment in the computer science major, which typically occurs during their freshman year. The difference in the means between the Fall 2018 course and the Spring 2019 and Fall 2019 courses was possibly due to a variance in the students' ability and/or their dispositions.

When the unit test scores for the two semesters that were included in the study were compared, the mean for the Unit 1 Test for Fall 2019 was 4.6 points lower than the mean for Spring 2019. The Unit 2 Test for Fall 2019 was 5.25 points higher than the mean for Spring 2019. The Unit 1 Test and the Unit 2 Test for both semesters were taught using a traditional lecture and lab-based learning format. The Unit 3 Test and the Unit 4 Test for Fall 2019 were taught using a team-based learning format. The Unit 3 Test and the Unit 4 Test for Spring 2019 were taught using the traditional lecture and lab-based learning format. The mean for the Unit 3 Test for Fall 2019 was 2.04 points higher than the mean for Spring 2019. The mean for the Unit 4 Test for Fall 2019 was 0.04 points higher than the mean for the Spring 2019 section. The descriptive statistics for the Spring 2019 and Fall 2019 semesters are presented in Table 7.

Table 7

Descriptive Statistics for the Unit Tests for the Spring 2019 and Fall 2019 Semesters

Semester	Test	*N*	*min*	*max*	*M*	*SD*
Spring 2019	Unit 1	28	16	98	66.11	23.97
	Unit 2	28	45	97	73.39	13.42
	Unit 3	28	47	110	86.82	14.60
	Unit 4	28	31	100	77.46	18.06
Fall 2019	Unit 1	24	32	97	70.71	19.76
	Unit 2	24	32	94	68.54	17.10
	Unit 3	24	52	106	88.86	12.74
	Unit 4	24	44	100	77.50	14.89

Although the findings in the review of literature showed significant improvements in the pass rate and retention rate of the course using team-based learning (Comeford, 2016; Lasserre & Szostak, 2011), the results of the current research did not find statistically significant differences in the numerical unit test scores between the two groups. The first research question sought to determine if there was a difference in the knowledge of conditional expressions as measured by unit test scores between the students who took the course using a traditional lecture and lab-based course and the students who took the course using a team-based learning course while controlling for the students' knowledge in programming terminology and data types. The results of the ANCOVA indicated that the difference in the means was not statistically significant; therefore, the researcher failed to reject the null hypothesis. The second research question sought to determine if there was a difference in the knowledge of looping expressions as measured by unit test scores between the students who took the course using a traditional lecture and lab-based course and the students who took the course using a team-based learning course while controlling for the students' knowledge in programming terminology and data types. The results of the ANCOVA indicated that the difference in the means was not statistically significant; therefore, the researcher failed to reject the null hypothesis.

Najdanovic-Visak (2017) only used team-based learning in one module of an engineering course and found that, even though the students liked working together in teams and getting to know their classmates better, they could have improved their scores and success in the course if more of the class content was taught using team-based learning. In the current study, the treatment was the first time that the instructor had taught the course using team-based learning. In addition, the participants in the study had never taken a course that implemented team-based learning. Changing the teaching methodology from a traditional lecture and lab-based teaching methodology to a team-based teaching methodology midway through the semester may have been more challenging for the students than if the students stayed with a consistent teaching methodology for the entire semester. With proper training for the instructor and the students and by teaching the entire class using team-based learning, the results may have been more consistent with the literature that was presented in Chapter II (e.g., Comeford, 2016; Jeno et al., 2017; Najdanovic-Visak, 2017).

The results from the current study indicated that no statistically significant differences in the unit test scores existed between the students who took the course using a team-based learning format as compared to the students who took the course using a traditional lecture and lab-based format. However, Remington et al. (2017) found that team-based learning improved the test scores of the students in an elective pharmacotherapeutics course as compared to lecture-based courses and gave the students more confidence in their ability to make decisions after taking the course using team-based learning. Similarly, Lasserre and Szostak (2011) found that team-based learning reflected major improvements in the withdrawal rate and the success rate of the students completing the introduction to programming course as measured by the final course grades of the participants in their study, which included the unit test scores and the other assignment and quiz scores.

Other benefits of using team-based learning that were not included in the current study but were found in previous studies included that the students who took a course using team-based learning valued the in-class collaboration with their team members, and most students believed that team-based learning was a good use of the time in the class (Kirkpatrick, 2017; Lykke, 2015). Walker and Zheng (2017) found that students perceived that they learned better using team-based learning and preferred team-based learning over the traditional lecture and lab-based instructional strategy. In addition, a critical part of adult learning is for the learner to know how to look for answers on his or her own by using past experiences to make future decisions (Pappas, 2013). Nikooravesh et al. (2016) determined that, by using team-based learning, the team members were able to think, create, and learn together, which sped up the teaching and learning process. Team members also learned from their experiences how to learn better and how to find new approaches while extending their own knowledge. Matalonga et al. (2017) found that students perceived that a course taught using team-based learning had more advantages than a course taught using a traditional lecture-based course, and the students perceived that team-based learning allowed them to use their own individual learning style. Pardamean et al. (2017) found that the students performed better when they worked in teams, and Jeno et al. (2017) concluded that using team-based learning improved the students' confidence in their ability to learn the course material. Further studies could consider these additional benefits. Frame et al. (2015) also suggested that universities should implement team-based learning early in the curriculum to allow students time to understand the benefits that team-based learning added to their education.

Limitations of the Study

The current study was conducted in a small university for a required course within certain majors; therefore, the results of the research may not be able to be generalized to larger universities or elective courses. The small pre-existing sample that was used in the study was also a limitation of the study. The semester used for the study was the first time that team-based learning was taught by the instructor and was the first time that the students were exposed to team-based learning. With continued usage, the instructor and the students could become more familiar with the instructional strategy, which could improve unit test scores of the students. Another limitation of the study was the unexpected COVID-19 pandemic that occurred at the midpoint of the Spring 2020 semester. Due to the pandemic, all university system institutions were required to complete the semester fully online. This transition from in-class instruction to fully online instruction prohibited the use of the standardized intervention procedures during the Spring 2020 introduction to programming course. Therefore, only the data from Fall 2019 (i.e., team-based learning) could be included in the study.

Recommendations for Future Research

One recommendation for future research would be to determine a better teaching strategy for the Unit 1 and Unit 2 materials. Because Unit 1 covered foundational concepts (i.e., basic programming terminology and data types), team-based learning may not be the best teaching strategy for the material covered. Unit 1 may benefit from the use of paired programming or some other form of cooperative learning (Sarpong et al., 2013; Wang & Hwang, 2017). Because Unit 2 (i.e., methods and behaviors) included other foundational concepts, including the use of methods in programming, this unit could be taught using team-based learning (Makalew & Pardamean, 2017). Implementing the team-based learning strategy earlier in the course may improve the outcomes of using team-based learning with Unit 3 (i.e., conditional expressions) and Unit 4 (i.e., looping expressions) because the students would be more familiar with the process and the expectations. Implementing team-based learning earlier in the course may help avoid confusion that may have occurred by switching teaching strategies during the semester. Comeford (2016) and Lasserre and Szostak (2011) completed their studies on a course that was taught using team-based learning the entire semester and found improved pass rates and withdrawal rates. However, Rezaee et al. (2016) and Wang and Hwang (2017) both completed their studies on a course that was taught using a traditional lecture and lab-based teaching strategy during the first half of the course and a team-based learning teaching strategy during the second half of the course. Both research teams found higher test scores with the sections that were taught using team-based learning.

Determining the students' opinions with the use of interviews or surveys could also be beneficial. Knowing the students' point of view of using team-based learning and what benefits that the students felt did or did not have on the students' ability to be successful in the course could help to improve the way that the course is taught (Lykke et al., 2015). The results of this research could also show other benefits of using team-based learning other than just an improvement in the students' unit test scores.

Starting the students in the course with an assignment that could build the students' self-concept could be beneficial. For most students, the introduction to programming course is the first course that is taken in their computer science major. The students might begin this course with the thought that they will not be successful in the course (Hegazi & Alhawarat, 2015). This preconceived notion could derive from hearing about other students' experiences in the course. Building the students' confidence in their ability to be successful in the course could help to improve their success in the course (Hegazi & Alhawarat, 2015).

Alturki (2016) noted that, when students fail or withdraw from the introduction to programming courses, the students fall behind in their coursework, which causes them to graduate later than originally planned. In addition, the students are more likely to change their majors to avoid having to retake the course. Sarpong et al. (2013) found that students perceived that using team-based learning was a better teaching strategy that could increase the participation of the students and could lead to a higher success rate in the introduction to programming course. Future research could determine the attrition rate in the computer science program based on the failure and withdrawal rates in the introduction to programming course according to the teaching strategies implemented within the course. Future research could determine the benefits of using team-based learning in other STEM fields. Team-based learning could be implemented in non-lab courses as well to help improve team building and communication skills that are needed for students to be successful in their future careers.

Implications of the Study

With proper training and by using team-based learning earlier in the introduction to programming course, a higher level of success in the course may be possible. Courses taught earlier in the curriculum could include team-based learning components to increase the students' exposure to team-based learning. Incorporating team-based learning in other courses could also improve the success rates in those courses and help to increase the communication skills of the students (Epsey, 2018). The improvement of the success rate could improve the attrition rate in the computer science major, which could improve the graduation rate in the degree program (Comeford, 2016). A higher graduation rate in the degree program could help to improve the problem of not having enough computer science graduates to fill the current and expected job demands (Bureau of Labor Statistics, 2015; Microsoft, 2012).

Faculty members within the computer science department should work together to develop team-based learning modules that would be appropriate for other courses in the computer science degree program. These modules could help to build the students' skills for working together in teams and being able to communicate with each other (Pardamean et al., 2017). By including team-based learning in more courses, the students' success in those courses could improve as the students become more familiar with the learning strategy (Frame et al., 2015). The team skills and communication skills taught using team-based learning could increase the success rates in the course (Kirkpatrick, 2017).

Conclusion

Although the research did not find statistically significant differences in the unit tests scores between the students who took the introduction to programming course using a traditional lecture and lab-based format and the students who took the introduction to programming course using a team-based learning format, there were other advantages to the students. The students who took the course using a team-based learning format were able to learn how to work together in teams and learn to adjust and overcome personality conflicts, which will benefit them in their future careers. The students also were able to get to know other students in their major, which may help them in future courses that they might take together. Another benefit was that the team-based learning course was able to move at a slightly faster pace because the students not only had the instructor to help them with their assignments, but they also had their fellow team members. The team-based learning course was able to cover an additional unit that the traditional lecture and lab-based course did not have time to complete. With more training for the instructor and the students, I believe that more improvements would be evident. If the success rate of the introduction to programming course could improve, the number of students who are retained in the computer science degree program could improve, which could lead to an increased graduation rate. More graduates in computer science could help fill the growing number of jobs in this field and could help fill the expected needs (Bureau of Labor Statistics, 2015; Microsoft, 2012).

REFERENCES

Aljohani, O. (2016). A review of the contemporary international literature on student retention in higher education. *International Journal of Education & Literacy Studies, 4*(1), 40-52. doi: 10.7575/aiac.ijels.v.4n.1p.40

Alturki, R. A. (2016). Measuring and improving student performance in an introductory programming course. *Informatics in Education, 15*(2), 183-204. doi: 10.15388/infedu.2016.10

Alvarez-Bell, R., Wirtz, D., & Bian, H. (2017). Identifying keys to success in innovative teaching: Student engagement and instructional practices as predictors of student learning in a course using a team-based learning approach. *Teaching & Learning Inquiry, 5*(2), 128-146. doi: 10.20343/teachlearninqu.5.2.10

Beaubouef, T., & Mason, J. (2005). Why the high attrition rate for computer science students: Some thoughts and observations. *ACM SIGSCE Bulletin, 37*(2), 103-106. doi: 10.1145/1083431.1083474

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *SIGCSE Bulletin*, *39*(2), 32–36. doi: 10.1145/1272848.1272879

Bennedsen, J., & Caspersen, M. E. (2019). Failure rates in introductory programming: 12 years later. *ACM Inroads*, *10*(2), 30-36. doi: 10.1145/3324888

Bosse, Y., & Gerosa, M. A. (2017). Why is programming so difficult to learn? Patterns of difficulties related to programming learning mid-stage. *ACM SIGSOFT Software Engineering Notes, 41*(6), 1–6. doi: 10.1145/3011286.3011301

Bureau of Labor Statistics, U.S. Department of Labor. (2019). *Occupational outlook handbook, 2019 Edition, Computer Programmers*. Washington, DC: Author.

Burgess, A., Bleasel, J., Haq, I., Roberts, C., Garsia, R., Robertson, T., & Mellis, C. (2017). Team-based learning (TBL) in the medical curriculum: Better than PBL? *BMC Medical Education, 17*(1), 243-254. doi: 10.1186/s12909-017-1068-z

Burke, A. (2019). Student retention models in higher education: A literature review. *College and University, 94*(2), 12-21. doi: 10.2202/1949-6605.1138

Canedo, E. D., Santos, G. A., & Leite, L. L. (2018). An assessment of the teaching-learning methodologies used in the programming courses at a Brazilian university. *Informatics in Education, 17*(1), 45–59. doi: 10.15388/infedu.2018.03

Celli, L. M., & Young, N. D. (2017). Contemporary pedagogy for the adult learning, *PUPIL: International Journal of Teaching, Education and Learning, 1*(1), 86-96. doi: 10.20319/pijtel.2017.11.8696

Charles Babbage Institute (2019). *Who was Charles Babbage?*. Retrieved from http://www.cbi.umn.edu/about/babbage.html

Chen, J. C. (2014). Teaching nontraditional adult students: Adult learning theories in practice. *Teaching in Higher Education, 19*(4), 406-418. doi: 10.1080/13562517.2013.860101

Chen, J. C. (2017). Nontraditional adult learners. *SAGE Open*, *7*(1), 1-12. doi: 10.1177/2158244017697161

Chen, X. (2015). STEM attrition among high-performing college students: Scope and potential causes. *Journal of Technology and Science Education, 5*(1), 41-59. doi: 10.3926/jotse.136

College. (n.d.a). *Curriculum sheets.* Retrieved October 29, 2019 from http://www.[redacted]/academics/curriculum.

College. (n.d.b). *Fact book 2019*. Retrieved February 22, 2020 from https://www.[redacted]/Assets/research/FactBook/Enrollment%20by%20Age.pdf.

College. (n.d.c). *Undergraduate bulletin (2018-2019).* Retrieved October 29, 2019 from https://www.[redacted]/Assets/RegistrarsOffice/bulletin/Current-Undergraduate.pdf.

Comeford, L. (2016). Team-based learning reduces attrition in a first-semester general chemistry course. *Journal of College Science Teaching, 46*(2), 42-46. doi: 10.2505/4/jcst16_046_02_42

Conner, L. R., Richardson, S., & Murphy, A. L. (2018). Using adult learning principles for evidence-based learning in a BSW research course. *Journal of Baccalaureate Social Work*, *23*(1), 355–365. doi: 10.18084/1084-7219.23.1.355

Dong, Y., Stupnisky, R. H., Obade, M., Gerszewski, T., & Ruthig, J. C. (2015). Value of college education mediating the predictive effects of causal attributions on academic success. *Social Psychology of Education, 18*(3), 531-546. doi: 10.1007/s11218-015-9299-5

Doyle, B. (2015). *C# programming, From problem analysis to program design* (5th ed.). Boston, MA: Course Technology.

Ede, A., & Cormack, L. B. (2017). *A history of science in society : From philosophy to utility, 3*. North York, Ontario: University of Toronto Press, Higher Education Division.

Emke, A. R., Butler, A. C., & Larsen, D. P. (2016). Effects of team-based learning on short-term and long-term retention of factual knowledge. *Medical Teacher, 38*, 306-311. doi: 10.3109/0142159X.2015.1034663

Espey, M. (2018). Diversity, effort, and cooperation in team-based learning. *The Journal of Economic Education, 49*(1), 8-21. doi: 10.1080/00220485.2017.1397571

Filliben, J. J., & Heckert, A. (n.d.). *NIST/SEMATECH e-Handbook of statistical methods*. Washington, DC: National Institute of Standards and Technology, U.S. Department of Commerce. Retrieved June 19, 2020 from https://doi.org/10.18434/M32189

Frame, T. R., Cailor, S. M., Gryka, R. J., Chen, A. M., Kiersma, M. E., & Sheppard, L. (2015). Student perceptions of team-based learning vs traditional lecture-based learning. *American Journal of Pharmaceutical Education, 79*(4), 1-11. doi: 10.5688/ajpe79451

Hammarberg, K., Kirkman, M., & de Lacey, S. (2016). Qualitative research methods: When to use them and how to judge them. *Human Reproduction, 31*(3), 498–501. doi: 10.1093/humrep/dev33

Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computers & Education*, *54*(4), 1127–1136. doi: 10.1016/j.compedu.2009.10.020

Hegazi, M. O., & Alhawarat, M. (2015). The challenges and the opportunities of teaching the introductory computer programming course: Case study. In J. E. Guerrero (Ed.), *Proceedings of the fifth international conference on e-Learning,* (pp. 324-330). Piscataway, NJ: Institute of Electrical and Electronics Engineers. doi: 10.1109/ECONF.2015.61

Horton, D. & Craig, M. (2015). Drop, fail, pass, continue: Persistence in CS1 and beyond in traditional and inverted delivery. In A. Decker & K. Eiselt (Chairs), *SIGCSE '15 Proceedings of the 46th ACM technical symposium on computer science education* (pp. 235-240). New York, NY: Association for Computing Machinery. doi: 10.1145/2676723.2677273

Huang, C., & Lin, C. (2017). Flipping business education: Transformative use of team-based learning in human resource management classrooms. *Educational Technology & Society, 20*(1), 323-336.

IBM. (n.d.). *IBM100 – The origins of computer science*. Retrieved September 15, 2019 from www.ibm.com/ibm/history/ibm100/us/en/icons/compsci/.

Ifrah, G. (2001). *The universal history of computing: From the abacus to the quantum computer.* New York, NY: John Wiley & Sons, Inc.

Jeno, L. M., Raaheim, A., Kristensen, S. M., Kristensen, K. D., Hole, T. N., Haugland, M. J., & Maeland, S. (2017). The relative effect of team-based learning on motivation and learning: A self-determination theory perspective. *CBE Life Sciences, 16*(4), 1-16. doi: 10.1187/cbe.17-03-0055

Johnson, R. B., & Christensen, L. (2017). *Educational research: Quantitative, qualitative, and mixed approaches.* (6th ed.). Los Angeles, CA: SAGE.

Johnson, S. N. (2017). Comparing student learning in the team-based learning classroom with different team reporting methods. *Journal of College Science Teaching, 47*(2), 76-82. doi: 10.2505/4/jcst17_047_02_76

Joint Task Force on Computing Curricula, Association for Computing Machinery and IEEE Computer Society. (2013). *Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science.* New York, NY: Association for Computing Machinery. doi: 10.1145/2534860

Kellenberg, F., Schmidt, J., & Werner, C. (2017). The adult learner: Self-determined, self-regulated, and reflective. *Signum Temporis, 9*(1), 23-29. doi: 10.1515/sigtem-2017-0001

Kirkpatrick, M. S. (2017). Student perspectives of team-based learning in a CS course: Summary of qualitative findings. In M. E. Caspersen & S. H. Edwards (Chairs), *SIGCSE '17: Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education* (pp. 327-332). Seattle, WA: Association for Computing Machinery. doi: 10.1145/3017680.3017699

Knowles, M. S. (1973). *The adult learner: A neglected species*. Houston, TX: Gulf Publishing Company.

Knowles, M. S. (1980). *The modern practice of adult education*. New York, NY: Cambridge.

Knowles, M. S. (1984). *Andragogy in action. Applying modern principles of adult education*. San Francisco, CA: Jossey Bass.

LaMorte, W. (2015, September 30). *Faculty resources for team-based learning*. Retrieved from http://sphweb.bumc.bu.edu/otlt/MPH-Modules/Teams/teams6.html.

Lasserre, P., & Szostak, C. (2011). Effects on team-based learning on a CS1 course. In G. RoBling, T. Naps, & C. Spannagel (Chairs), *ITiCSE '11: Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (pp. 133-137). New York: NY: Association for Computing Machinery. doi: 10.1145/1999747.1999787

Levene, H. (1960). Robust tests for equality of variances. In I. Olkin et al. (Eds.). *Contributions to probability and statistics: Essays in honor of Harold Hotelling* (pp. 278-292). Stanford, CA: Stanford University Press.

Lomax, R. G., & Hahs-Vaughn, D. L. (2012). *An introduction to statistical concepts*. New York, NY: Psychology Press.

Lykke, M., Coto, M., Jantzen, C., Mora, S., & Vandel, N. (2015). Motivating students through positive learning experiences: A comparison of three learning designs for computer programming courses. *Journal of Problem Based Learning in Higher Education, 3*(2), 80-108. doi: 10.5278/ojs.jpblhe.v0i0.1130

MacVarish, K., & Cox, H. (2015, September 30). *Faculty resources for team-based learning*. Retrieved from http://sphweb.bumc.bu.edu/otlt/MPH-Modules/Teams/teams6.html.

Makalew, B. A. & Pardamean, B. (2017). Team based learning in computer science students. In L. S. Riza, A. Pranolo, A. P. Wibawa, E. Junaeti, Y. Wihardi, U. Hashim, S. Horng, R. Drezewski, H. S. Lim, G. Chakraborty, L. Hernandez, & S. Nazir (Eds.), *Proceedings of the 2017 3rd international conference on science in information technology (ICSITech)* (pp. 296-301). New York, NY: Institute of Electrical and Electronics Engineers. doi: 10.1109/icsitech.2017.8257128

Malik, K., & Khaliq, T. (2017). Andragogy (Adult learning). *The Annals of Pakistan Institute of Medical Sciences, 13*(4), 272-274.

Matalonga, S., Mousqués, G., & Bia, A. (2017). Deploying team-based learning at undergraduate software engineering courses. In R. Balof (Chair), *2017 IEEE/ACM 1st international workshop on software engineering curricula for millennials* (pp.

9-15). Piscataway, NJ: Institute of Electrical and Electronics Engineers. doi: 10.1109/SECM.2017.2

Microsoft Corporation. (2012). A national talent strategy: Ideas for securing U.S. competitiveness and economic growth. Retrieved from http://www.microsoft.com/en-us/news/download/presskits/citizenship/MSNTS.pdf.

Michaelsen, L., Knight, A. B., & Fink, L. D. (2004). *Team-based learning: A transformative use of small groups in college teaching.* Sterling, VA: Stylus Publishing.

Muduli, A., Kaura, V., & Quazi, A. (2018). Pedagogy or andragogy? Views of Indian postgraduate business students. *IIMB Management Review, 30*, 168-178. doi: 10.1016/j.iimb.2018.01.008

Najdanovic-Visak, V. (2017). Team-based learning for first year engineering students. *Education for Chemical Engineers, 18*, 26-34. doi: 10.1016/j.ece.2016.09.001

National Center for Education Statistics. (2009). *Digest of Educational Statistics: 2009*. (NCES Publication No. 2010-013). Washington, DC: Author. Retrieved from http://nces.ed.gov/programs/digest/d09/tables/dt09_192.asp?referrer=list

Nikooravesh A., Parpoochi, A., & Davoudi, A. H. M. (2016). The impact of team-based learning on knowledge development. *International Journal of Organizational Leadership, 5*(2), 110-122. doi: 10.33844/ijol.2016.60400

Pappas, C. (2013, May 9). *The adult learning theory – Andragogy – of Malcolm Knowles*. Retrieved from https://elearningindustry.com/the-adult-learning-theory-andragogy-of-malcolm-knowles.

Pardamean, B., Muljo, H. H., Suparyanto, T., Masli, E., Prabowo, H., & Donovan, J. D. (2017). Team-based learning as an instructional strategy: A comparative study. *The New Educational Review*, *4*(50), 134-145.

Patten, M. L., & Newhart, M. (2018). *Understanding research methods: An overview of the essentials* (10th ed.). New York, NY: Routledge.

Peterson, A., Craig, M., Campbell, J., & Tafliovich, A. (2016), Revisiting why students drop CS1. In J. Sheard & C. S. Montero (Chairs), *Koli calling '16: Proceedings of the 16th Koli calling international conference on computing education research* (pp. 71-80). New York, NY: Association for Computing Machinery. doi: 10.1145/2999541.2999552

Quille, K., & Bergin, S. (2019). CS1: How will they do? How can we help? A decade of research and practice. *Computer Science Education, 29*(2/3), 254-282. doi: 10.1080/08993408.2019.1612679.

Rau, P. P., Xie, A., Li, Z., & Chen, C. (2016). The cognitive process of Chinese abacus arithmetic. *International Journal of Science and Math Education, 14*, 1499–1516. doi: 10.1007/s10763-015-9658-x

Remington, T. L., Bleske, B. E., Bartholomew, T., Dorsch, M. P., Guthrie, S. K., Klein, K. C., Tingen, J. M., & Wells, T. D. (2017). Qualitative analysis of student perceptions comparing team-based learning and traditional lecture in a pharmacotherapeutics course. *American Journal of Pharmaceutical Education, 81*(3), 1-10.

Rezaee, R., Moadeb, N., & Shokrpour, N. (2016). Team-based learning: A new approach toward improving education. *Acta Medica Iranica, 54*(10), 678-682.

Rosen S., & Rice J. R. (1994). The origins of computing and computer science at Purdue University. In J. Rice & R. A. DeMillo (Eds.). *Studies in computer science, software science, and engineering* (pp. 31-44). Boston, MA: Springer. https://doi.org/10.1007/978-1-4615-1791-7_5

Ross-Gordon, J. (2011). Research on adult learners: Supporting the needs of a student population that is no longer nontraditional. *Peer Review, 13*(1), 26-29.

Salkind, N. J. (2010). *Encyclopedia of research design*. Thousand Oaks, CA: SAGE Publications. doi: 10.4135/9781412961288

Sarpong, K. A., Arthur, J. K., & Amoako, P. Y. O. (2013). Causes of failure of students in computer programming courses: The teacher-learner perspective. *International Journal of Computer Applications, 77*(12), 27-32. doi: 10.5120/13448-1311

Sogunro, O. A. (2015). Motivating factors for adult learners in higher education. *International Journal of Higher Education, 4*(1), 22-37. doi: 10.5430/ijhe.v4n1p22

Stein, R. E., Colyer, C. J., & Manning, J. (2016). Student accountability in team-based learning classes. *Teaching Sociology, 44*(1), 28-38. doi: 10.1177/0092055X15603429

Sultana, S. P., & Reed, G. A. (2017). Curriculum for an introductory computer science course: Identifying recommendations from academia and industry. *Journal of Technology Studies, 43*(2), 80-92. doi: 10.21061/jots.v43i2.a.3

Smithsonian Science Education Center. (n.d.). *The STEM imperative*. Retrieved March 25, 2016 from https://ssec.si.edu/stem-imperative.

Team-Based Learning Collaborative (n.d.). *TBL consultant – Larry Michaelsen*. Retrieved November 30, 2020 from http://www.teambasedlearning.org/talk-to-the-experts/larry-michaelsen/

U.S. Department of Education, National Center for Education Statistics. (2019). *Digest of Education Statistics, 2017* (NCES Publication No. 2018-070). Washington, DC: Author. Retrieved from https://nces.ed.gov/pubs2018/2018070.pdf

Vihavainen, A., Airaksinen, J., & Watson C. (2014). A systematic review of approaches for teaching introductory programming and their influence on success. In Q. Cutts, B. Simon, & B. Dorn (Chairs), *ICER '14 Proceedings of the tenth annual conference on International computing education research (pp. 19-26).* New York, NY: Association for Computing Machinery. doi: 10.1145/2632320.2632349

Walker, Z. M., & Zheng, T. G. (2017). Adopting team-based learning for in-service teachers: A case study. *International Journal for the Scholarship of Teaching and Learning, 11*(1), Article 6.

Wang, X., & Hwang, G. (2017). A problem posing-based practicing strategy for facilitating students' computer programming skills in the team-based learning mode. *Association for Educational Communications and Technology*, *65*(6), 1655-1671. doi: 10.1007/s11423-017-9551-0

Watson, C., & Li, F. W. B. (2014). Failure rates in introductory programming revisited. *Proceedings of the ITiCSE, ACM, Uppsala, Sweden*, 39-44.

Willans, J., & Seary, K. (2018). "Why did we lose them and what could we have done"? *Student Success, 9*(1), 47-60. doi: 10.5204/ssj.v9i1.432

Wu, S., Farquhar, J., & Compton, S. (2018). Why do team-based-learning educators use TBL? *The Asia-Pacific Scholar, 3*(1), 38-41.

Yoon, H. B., Park, W. B., Myung, S., Moon, S. H., & Park, J. (2018). Validity and reliability assessment of a peer evaluation method in team-based learning classes. *Korean Journal of Medical Education, 30*(1), 23-29. doi: 10.3946/kjme.2018.78.

APPENDICES

Appendix A

Assessment of Individual Team Member's Team Working Skills

Rubric for Assessment of Individual Team Member's Team Working Skills

Background: A major part of this course involves team work and a team project with a final presentation. The team project plays an important role in further developing the team working skills of the students. The rubric below is used to evaluate each student with respect to his/her ability to function on teams.

Rubric: Each student is evaluated by team members along three dimensions, described below. Each of these dimensions is assigned a score of 1 through 4, with values representing increasing degrees of achievement in the particular dimension. The last column is the actual points assigned to this particular student, based on his or her actual performance. The overall total score is assigned by adding together the points corresponding to the three dimensions.

Name of person being evaluated: ________________________ **Team:** ________________________

Evaluator's name: ________________________ **Date of evaluation:** ________________________

Dimension	Score 1	Score 2	Score 3	Score 4	Points
1. Contribution to the team project/work	• Does not collect any relevant information • No useful suggestions to address team's needs	• Collects information when prodded • Tries to offer some ideas, but not well developed and/or clearly expressed to meet team's needs	• Collects basic, useful information related to the project • Occasionally offers useful ideas to meet the team's needs	• Collects and presents to the team a great deal of relevant information • Offers well-developed and clearly expressed ideas directly related to the group's purpose	
2. Taking responsibility	• Does not perform assigned tasks • Often misses meetings and, when present, does not have anything constructive to say • Relies on others to do the work	• Performs assigned tasks but needs many reminders • Attends meetings regularly but generally does not say anything constructive • Sometimes expects others to do his/her work	• Performs all assigned tasks • Attends meetings regularly and usually participates effectively • Generally reliable	• Performs all tasks very effectively • Attends all meetings and participates enthusiastically • Very reliable.	
3. Valuing other team members and quality of interactions	• Often argues with team mates • Doesn't let anyone else talk • Occasional personal attacks and "put-downs" • Wants to have things done his/her way and/or does not listen to alternate approaches	• Usually does much of the talking • Does not pay much attention when others talk • Often assumes others' ideas will not work • No personal attacks and put-downs but sometimes patronizing	• Generally listens to others' points of view • Always uses appropriate and respectful language • Tries to make a definite effort to understand others' ideas	• Always listens to others and their ideas • Helps them develop their ideas while giving them full credit • Always helps the team reach a fair decision	
				Total:	

Comments (optional):

Appendix B

Assessment of Team Effectiveness

Rubric for Assessment of Team Effectiveness

Background: A major part of this course involves team work. Individual contributions impact the team's effectiveness and separately, you will be assessing your team members. This rubric allows your team to collectively, and honestly, evaluate how your team performed throughout the week. Consider your team's work in each category described below. Each of these categories is assigned a score of 1 through 3, with values representing increasing degrees of achievement. The last column is the actual points assigned to this particular team. The overall total score is assigned by adding together the points corresponding to the categories. The professors will contribute to the final score, based on their observations of the team during the week.

Name of Team: ______________________________ **Date:** __________

Category	Score 1	Score 2	Score 3	Points
1. Attitude and Team Climate	• Complacent energy level • Disrespectful or inattentive attitudes displayed • Conflicts were common and/or unresolved	• Generally upbeat energy level • Attentive and polite attitudes displayed often • Conflicts, if any, were defused	• Inspiring and motivating energy level • Attentive and polite attitudes displayed always • Conflicts, if any, were resolved and helped the team grow	
2. Team's In-class Participation	• Uneven • Quality was inconsistent and/or lackluster	• Moderate • Quality was good	• Extensive • Quality was excellent	
3. Team's operating procedures	• Did not follow instructions • Did not meet deadlines • Could not depend on each other	• Followed most instructions • Met most deadlines • Could depend on each other frequently, but not always	• Followed all instructions • Met all deadlines • Could always depend on each other	
4. Team's self-motivation and professionalism	• One or more team members were generally late for class, did not adhere to breaks and/or left before class was dismissed	• One or more team members were occasionally late for class, did not adhere to breaks, and left before class was dismissed	• All team members were on time for class, adhered to breaks, and stayed until class was dismissed	
5. Synergy and Team Cohesiveness	• Team was a collection of individuals that merely divided the work	• Team realized some benefit from working together beyond a simple division of labor	• Team attained a high level of synergy and developed skills and ideas through interactions with each other	
TOTAL POINTS				

Appendix C

Group Project Evaluation

Group Project Evaluation

Your Name: ______________________ Group Project Title: ______________________

Please rate the following:

1) The overall quality of your project?	Excellent	Good	Fair	Poor
2) The overall cooperation among team members?	Excellent	Good	Fair	Poor
3) The educational value of the project?	Excellent	Good	Fair	Poor
4) Your personal contribution to the intellectual content?	Excellent	Good	Fair	Poor

5) List and describe the specific contributions you made to your group's project.

6) How would you rate your contributions in comparison to that of other group members?

a) I contributed more than most of the other members.

b) I contributed less than most of the other members.

c) I contributed an average amount compared to the others.

d) All group members contributed more or less equally.

Your Comments:

7) How would you describe the equity and distribution of work among the members of the group?

8) Were any members of your group overly domineering?

9) What are your thoughts regarding the value of the team project with respect to "cross-cutting" competencies such as communication skills, time management, team work, and accountability?

Appendix D

Approval to Use Group Evaluation Assessment

COLUMBUS STATE UNIVERSITY

Karen Cook [Student] <cook_karen3@columbusstate.edu>

TBL Group Project Evaluation use request

Lamorte, Wayne W <wlamorte@bu.edu> Sun, Jun 7, 2020 at 2:20 PM
To: "Karen Cook [Student]" <cook_karen3@columbusstate.edu>

Hi Karen,

Yes, you have my permission to use that form. Good luck with your project.

Wayne LaMorte

> On Jun 7, 2020, at 12:21 PM, Karen Cook [Student] <cook_karen3@columbusstate.edu> wrote:
>
> Good afternoon Dr. LaMorte,
>
> My name is Karen Cook. I am a current Ed.D. student at Columbus State University in Columbus, GA. I am also a Senior Lecturer at Georgia Southwestern State University in Americus, GA. I am the programming instructor at the University and have recently implemented team-based learning in my courses. My dissertation research involves determining if team-based learning will improve the attrition rates of the students in the computer science and information technology majors. I found the group project evaluation that you created on the Faculty Resources for Team-Based Learning website (http://sphweb.bumc.bu.edu/otlt/MPH-Modules/Teams/teams6.html). I am writing to ask your permission to use the form in my course and in my research and will reference your contribution and the website where the form is located.
>
> Thank you for your consideration,
> Karen Cook

Appendix E

Approval to Use Rubric for Assessment of Team Effectiveness and the Rubric for Assessment of Individual Team Members

COLUMBUS STATE UNIVERSITY

Karen Cook [Student] <cook_karen3@columbusstate.edu>

Request to use TBL rubrics

Macvarish, Kathleen <kmacvar@bu.edu> Mon, Jun 8, 2020 at 8:38 AM
To: "Cox, Harold D" <hcox@bu.edu>, "Karen Cook [Student]" <cook_karen3@columbusstate.edu>

Hi Karen,

It's fine with me if you use (and adapt) the rubrics– please note that I used a variety of sources to pull them together so they're not entirely my original work.

Kathleen

Kathleen MacVarish

Associate Professor of the Practice

Boston University School of Public Health

715 Albany Street

Boston, MA 02118

617-358-4076

Appendix F

IRB Approval

Institutional Review Board
Columbus State University

Date: 8/20/20
Protocol Number: 21-003
Protocol Title: The Use of Team-Based Learning to Improve Success Rates of Adult Learners in Introduction to Programming Courses in a Small Southwest Georgia University
Principal Investigator: Karen Cook
Co-Principal Investigator: Jennifer Brown

Dear Karen Cook:

The Columbus State University Institutional Review Board or representative(s) has reviewed your research proposal identified above. It has been determined that the project is classified as exempt under 45 CFR 46.101(b) of the federal regulations and has been approved. You may begin your research project immediately.

Please note any changes to the protocol must be submitted in writing to the IRB before implementing the change(s). Any adverse events, unexpected problems, and/or incidents that involve risks to participants and/or others must be reported to the Institutional Review Board at irb@columbusstate.edu or (706) 507-8634.

If you have further questions, please feel free to contact the IRB.

Sincerely,

Amber Dees, IRB Coordinator

Institutional Review Board
Columbus State University