TEACHING PEDAGOGY AND
PREFERRED LEARNING STYLES FOR INTRODUCTORY COMPUTER
PROGRAMMING COURSES

A Dissertation
presented to
the Faculty of the Graduate School
at the University of Missouri - Columbia

by
CARLA BATES
Dr. Joi Moore, Dissertation Advisor
July 2017

The undersigned, appointed by the dean of the Graduate School, have examined the
dissertation entitled

TEACHING PEDAGOGY AND
PREFERRED LEARNING STYLES FOR INTRODUCTORY COMPUTER
PROGRAMMING COURSES

presented by Carla Bates,

a candidate for the degree of doctor of philosophy,

and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Joi Moore

Dr. Sanda Erdelez

Dr. Dale Musser

Dr. Johannes Strobel

Thanks to my dad Clarence for not letting me give up.

Thanks to my daughters Danielle and Samantha for their patience with me going to school while working a full time job and multiple part-time jobs at a time while you were growing up.

Love you all!

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Joi Moore and members of my committee: Dr. Sanda Erdelez, Dr. Dale Musser, Dr. Johannes Strobel. I would also like to thank Dr. Fiona Nah and Dr. Nathan Twyman for feedback and Dr. Mike Hilgers for feedback and editing. In addition, I would like to thank Dr. Keng Siau for his patience with my assignments at work while finishing my dissertation. Thanks to Sam Smith for helping me with collection and organization of the data.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	viii
ABSTRACT	ix
Chapter One -- Introduction.....	1
Computer Programming	1
Problem Solving Versus Programming	2
Preferred Teaching and Learning Modes.....	4
Self-Regulated Learning	6
Investigating Modern Learning Theory in the Computer Programming Classroom.....	7
Research Questions	8
Summary	9
DEFINITIONS.....	11
Chapter Two - Literature Review	13
Teaching/Learning Modes	13
Dunn and Dunn Model	16
Felder and Silverman Learning Style	17
Gregorc Learning Style Model	17
Kolb Experiential Learning Model	18
Revised Approaches to Studying Inventory (RASI) Model	19
Visual, Aural, Read-Write, Kinesthetic (VARK®) Learning Model	19
Measuring Preferred Learning Modes - VARK®	20
Practicality of Teaching and Learning Modes	26
Basic Computer Programming Skills	27
Barriers Programmers Encounter	28
Worked Examples	30
Self-Regulated Learning	31
Motivated Strategies for Learning Questionnaire (MSLQ)	35
Summary	36
Chapter Three -- Methodology	37
Overview of Chapter Three	37
Research Questions	37
Instructional Environment	38
Overview of weekly activities	38
Visual	40
Aural	41
Read/Write	42
Kinesthetics	42
Summary of Materials	43
Participants.....	43
Procedure	44
IRB-approved informed consent form.....	44
Demographic information.....	44

VARK® Survey.....	44
Teaching and Learning Mode Choices	45
Assignments.....	47
Post-Assignment Survey.....	48
Additional Contact.....	48
Statistical Data	49
Research Question One.....	51
Research Question Two	51
Research Question Three	52
Assumptions.....	53
Pilot Study.....	54
Procedure	54
Qualitative Comments from Post Survey	55
Lessons Learned	55
Summary of Pilot Study Results.....	56
Chapter Four – Results.....	57
Summary of Chapter Four	57
Purpose of Study	57
Demographic Data	58
Learning Modes	59
Students’ Order of Choosing Teaching Modes.....	61
Students’ Rank of Teaching Modes Compared with Preferred Learning Mode	65
Assignment Scores by Preferred Learning Modes	78
Preferred Learning Modes by Choice of Venue for Example Program	81
Assignment Scores by Venue Used to Complete Example Assignment	84
Self-Regulated Score by Venue	88
Working on Assignments.....	93
Qualitative Analysis.....	95
Flowchart Qualitative Data.....	96
Audio Qualitative Data	96
Text Qualitative Data.....	97
Example Assignment - Qualitative Data	98
Choice of Out of Class Venue for Example Assignment	99
Watched Example More than Once	100
Chapter Five – Discussion	101
Summary of Chapter Five.....	101
Students’ Ranking of Teaching Modes (Preferred and Helpful)	101
Assignment Scores by Preferred Learning Modes	103
Assignment Scores by Venue Used to Complete Example Assignment	105
Venue for Completing Example Assignment by Preferred Learning Mode	107
Self-Regulated Score by Venue	107
Problems with Learning Styles and Assessment	108
Implications	110
Programming Courses	110
Instructional Design.....	113

Learning Styles	114
Limitations	115
Content of Additional Teaching Mode Materials	116
Venue of Completing Example Assignment	117
Prior Experience	118
Delimitations.....	118
Future Research	118
Conclusions.....	120
Appendix A: Course Syllabus.....	127
Appendix B: Consent Form	134
Appendix C: Pre-Questionnaire.....	135
Appendix D: VARK Questionnaire	139
Appendix E: Assignments and Rubrics	142
Summary	142
Assignment 8 Description.....	143
Assignment 8 Rubric	145
Assignment 8 Lecture Slides	148
Assignment 8 Flow Chart	153
Assignment 8 Text	154
Assignment 8 Audio Transcription.....	157
Assignment 8 Example Program	159
Assignment 8 Program Solution	161
Assignment 9 Rubric	165
Assignment 9 Lecture Slides	168
Assignment 9 Flow Chart	170
Assignment 9 Text	171
Assignment 9 Audio Transcript.....	179
Assignment 9 Example Program	182
Assignment 9 Program Solution	185
Appendix F: Post Assignment Questionnaire	187
Appendix H: Screen Shots Showing the Order of Teaching Mode	189
Appendix I: Tegrity (Example Assignment) Video.....	190
Appendix J: VARK® Survey Results.....	191
Vita.....	192

LIST OF TABLES

Table 1. Learning Model descriptions	16
Table 2: Phase Structure and Sub Processes of Self-Regulation Cyclical Self-Regulatory Phases.....	33
Table 3. Teaching-Learning Mode	39
Table 4. Research Questions, Data Collection Methods, and Analysis Used.....	49
Table 5: Gender of Participants by Preferred Learning Mode.....	61
Table 6: Chi-Square Tests. Gender by Preferred Learning Mode	61
Table 5: Assignment 8. VARK® Mode vs. First Mode Accessed	63
Table 6: Assignment 8. Chi-Square Statistics. VARK® Mode vs. Mode First Accessed	63
Table 7: Assignment 9. VARK® Mode vs. First Mode Accessed	64
Table 8: Assignment 9. Chi-Square Statistics. VARK® Mode vs. Mode First Accessed	64
Table 9: Summary of Most Helpful and Preferred Rankings by Students	66
Table 10: Student Ratings of Teaching Modes Material Compared to Preferred Learning Modes.....	66
Table 11: Group Statistics. Assignment 8 Scores. Mode Ranked Most Preferred vs. VARK® Preferred Mode	67
Table 12: Group Statistics. Assignment 8 Scores. Mode Ranked Most Helpful vs. VARK® Preferred Mode	68
Table 13: T-Test: Assignment 8 Scores. Mode Ranked Preferred vs. VARK® Preferred Mode	70
Table 14: T-Test: Assignment 8 Scores. Mode Ranked Most Helpful vs. VARK® Preferred Mode	72
Table 15: Group Statistics. Assignment 9 Scores. Mode Ranked Preferred vs. VARK® Preferred Mode	73
Table 16: Group Statistics. Assignment 9 Scores. Mode Ranked Most Helpful vs. VARK® Preferred Mode	74
Table 17: T-Test: Assignment 9 Scores. Mode Ranked Preferred vs. VARK® Preferred Mode	75
Table 18: T-Test: Assignment 9 Scores. Mode Ranked Most Helpful vs. VARK® Preferred Mode	77
Table 19. Assignment 8 Scores by Learning Mode.....	78
Table 20. Assignment 9 Scores by Learning Modes	79
Table 21. ANOVA Table for Assignment 8 Scores	80
Table 22. ANOVA Table for Assignment 9 Scores	80
Table 23. Example Assignment 8 Data.....	81
Table 24. Example Assignment 9 Data.....	82
Table 25: Chi-Square. Example Assignment 8. Preferred Mode * Venue	83
Table 26: Chi-Square. Assignment 9. Preferred Mode * Venue	84
Table 27. Assignment 8 Statistics. Example Venue * Assignment Score	85
Table 28. Assignment 8 t-test Calculations. Example Venue * Assignment Score	86
Table 29. Assignment 9 Statistics. Example Venue * Assignment Score	87
Table 30. Assignment 9 t-test calculations. Example Venue * Assignment Score	88

Table 31: Assignment 8 Group Statistics. MLSQ Score * Venue of Completing Example	89
Table 32. Assignment 8 t-test. MLSQ * Venue for Example	90
Table 33. Assignment 9 Group Statistics. MLSQ * Venue of Completing Example.....	90
Table 34. Assignment 9 t-test. MLSQ * Venue for Example.....	92
Table 35: Assignment 8. Number of Materials Used While Working on Assignment	93
Table 36: Assignment 8. VARK® Mode * Material Used While Working on Assignment	94
Table 37: Assignment 9. Number of Materials Used While Working on Assignment	94
Table 38: Assignment 9. VARK® Mode * Material Used While Working on Assignment	95
Table 39: Themes. Flow Chart Qualitative Comments	96
Table 40: Themes. Audio Qualitative Comments.....	97
Table 41: Themes. Text Qualitative Comments	98
Table 42: Themes. Example Assignment Qualitative Comments	98
Table 43: Themes. Choice of On-Line Venue for Example Assignment	99
Table 44: Themes. Reason for Watching Example More than Once	100

LIST OF FIGURES

Figure 1: Six Prominent Learning Styles.....	14
Figure 2. Students by preferred learning mode.....	25
Figure 3. Social Cognitive Model of Learning	32
Figure 4. Triadic Forms of Self-Regulation.....	32
Figure 5: Ages of Participants.....	58
Figure 6: Gender of Participants	59
Figure 7: Population by Preferred Learning Mode (VARK® survey)	60

ABSTRACT

As society moves into the information age, it has generated the need for people capable of computational, and computer-oriented tasks. Universities have responded with more disciplines requiring at least one programming course, but educators stand witness to students of all degree majors struggling in these introductory programming courses. Traditionally, these courses follow a classic pattern of lecture followed by homework programs written during lab sessions. This raises the question of whether students would be more likely to complete a beginning programming course successfully if instructors included various pedagogy styles.

In this research, the question is addressed by interacting with students through various methods in a controlled fashion to quantify the impact of student learning styles. This is achieved by the instructor posting programming teaching materials matching known learning style featuring variation in visual, aural, read/write, and kinesthetic (VARK¹) concentrations. Then the students complete the VARK®, A Guide to Learning Styles, questionnaire. It is found that there are not any significant statistical relationships in regard to preferred learning styles, but additional teaching material for different preferred learning styles were popular with the students.

¹ VARK is an abbreviation of visual, aural, read/write, and kinesthetic models of learning. VARK® is the copyright protected survey used in education research. The ® will be used to distinguish learning model from questioner.

Chapter One -- Introduction

Computer Programming

Although learning to program can be difficult, introductory programming courses have grown rapidly in recent years. Programming courses have increased because there are more occupations that include information technology services (Csorny, 2013). Programming courses used to only be in computer science curriculum, but now they are included in computer engineering, information systems, and information science programs (Pears et al., 2007). Students who are not computer science majors have a lack of motivation causing the learning process to fail (Forte & Guzdial, 2005). A beginning programming course that is designed for a variety of students' interests and backgrounds can motivate the non-computer science majors and lead them to less anxiety about the course (Bosch, D'Mello, & Mills, 2011; Brosnan, Gallop, Iftikhar, & Keogh, 2010; Forte & Guzdial, 2005; Gilroy & Desai, 1986; Igbaria & Chakrabarti, 2007). In a study conducted by Chang (2005), a correlation was found between computer anxiety and user perception of task complexity. In this study, only 98 of the 307 participants were computer science majors. All students were in an introductory course. Lack of computer confidence is likely to lead to computer anxiety, and students that have more anxiety about programming may see the programming tasks as more complex than they actually are (Chang, 2005). If the goal is too hard to reach or fails to heighten short-term motivation, initiative and interest will be reduced (Gao, 2011).

A static example in a textbook does not give the students insight into how the problem can be broken down and solved (Caspersen & Bennedsen, 2007). The programmer must not only design a mental model of the problem and solution, but also develop a design/model of the program and how it will run. There also needs to be a clear distinction between the model of what the program actually does and its intended purposes (Robins, Rountree, & Rountree, 2003). To achieve the goal of educating students in the programming field, educators are encouraged to use a framework to intertwine instruction of technical and non-technical skills, and the professional skills listed previously. This paradigm goes hand-in-hand with the two parts of programming described by Dou and Shi (2015): theory, and practice. The theory is explained during the input process, how the technique is supposed to work and be used or the logical thought behind the technique. The process “works out” the practice portion, and answers questions, such as: how does this actually work? When do I want to use this technique? Or, how do I implement the logic?

Problem Solving Versus Programming

Computer programming is a challenging course to teach in a way so that upon completion, students have achieved a level of mastery capable of consistently writing realistic programs (Lahtinen, Ala-Mutka, & Harvinen, 2005; Ramalingam, LaBelle, & Wiedenbeck, 2004; Robins et al., 2003; Rountree, Rountree, Robins, & Hannah, 2004). Writing a program is similar in nature to solving mathematics problems (Winslow, 1996). In particular, a programming assignment bears a strong resemblance to the so-called “word problems” which are considered by mathematic teachers to be among the hardest

for students to solve. Furthermore, Winslow (1996) argued that even though student programmers know how to solve a problem by hand, they have trouble converting the solution into a computer program.

Problem solving from a programming perspective requires students to complete four steps:

1. Understand the problem.
2. Solve the problem in two ways: (1) a mixture of paper and pencil calculations and mental jump work which solves the problem but does not describe how to solve it, and (2) a formal algorithm that can solve the problem given any input.
3. Translate the algorithm into the proper computer language.
4. Test and debug the program.

In their study, Lahtinen et al. (2005) found that students' biggest difficulties are not in understanding the concepts but in applying the concepts to solve the problem. In their study, they also found that students and teachers both agree that the most useful teaching materials are examples, and students also feel that the best way of learning is by doing. Jonassen (2004) suggested that practicing the skill being learned, or modeling performance, is an essential part of instruction. However, it is hard to create a curriculum that keeps everyone challenged and interested (Robins et al., 2003).

In short, different people learn in different ways. In education, the written word is the most common method to convey knowledge, through reading textbooks. How is it found out if the knowledge has transferred? Usually, in the past, knowledge is confirmed with a written test. Most people tend to cater to their preferred learning modes, Neil D Fleming (1995), also suggests that teachers may be reinforcing their own preference.

Even though using written mode is the traditional way in which it has always been done, it does not mean instructors should continue to travel down that path. It is hoped that once instructors realize “written-word teaching” may be hindering their students’ learning, they may modify their pedagogy to include the preferred learning modes of their students in a programming class.

Preferred Teaching and Learning Modes

To make matters even more difficult for instructors, students have different preferred teaching and/or learning modes (Neil D Fleming, 1995; Lujan & DiCarlo, 2006; Pears et al., 2007; Rosenberg & Burkert, 2015). Discussions about these different preferred learning modes have been around for many years and continue to be revised through the years. Pintrich, McKeachie, and Lin (1987) created a course to teach students how to learn including several different teaching modes and Pintrich (1991); Pintrich and De Groot (1990) researched different teaching modes in regard to self-regulated learning. Pintrich et al. (1987), and Reed and Bolstad (1991) compared teaching modes of methods versus examples and research using video as a teaching method (Sharp & Schultz, 2013; Whatley & Ahmad, 2007). A. Y. Kolb and Kolb (2005) considered several different teaching modes, and A. Y. Kolb and Kolb (2008) researched

different teaching modes via distance courses. These researchers were concentrating on a more specific type of teaching mode.

D. Kolb (1985) discusses Learning Styles Inventory, which identifies two dimensions of learning, perceiving (our preferred means of receiving new information) and processing (how we make sense of things) that occur in four phases: concrete experience (feeling), reflective observation (reflection), abstract conceptualization (thinking), and active experimentation (doing); and four quadrants or learning styles: accommodators (activists who learn best by becoming fully involved), convergers (pragmatists who are interested in the practical application of ideas), assimilators (theorists who enjoy working with ideas and models), and divergers (reflective learners who learn by observing and making sense of experiences).

A more updated and recent preferred learning mode evaluation includes visual, aural, reading/writing, and kinesthetic modes (Neil D Fleming, 1995). Visual mode students prefer to learn with graphs, charts, and flow diagrams and can work easily with symbols. Aural mode students learn better by ear, while reading/writing mode students prefer written word ways of learning. Kinesthetic mode students prefer concrete, multi-sensory experiences to learn through application. Fleming feels that the tertiary education should feel sorry for these kinesthetic mode students since these hands-on methods are not as prevalent in this level of education. He also admits that not all students and teachers are limited to one mode, but in some there will be a dominant preference or even an absence of some preferences.

Self-Regulated Learning

Preferred learning mode may not be the main driving force behind a student's preference for venue of learning. Barry J Zimmerman (1986) defines self-regulated learning as actions toward acquiring information or skill that involve goals. In other words, students initiate and direct their own efforts for knowledge and skills. Not only is there a social cognitive approach to self-regulated learning, but also three important aspects to educational psychologists:

1. It can explain the advantages of self-regulatory influences and behavioral ones while explaining the advantage of each
2. It can explain the impact of self-regulatory processes with social learning/behavioral experiences
3. Self-efficacy and strategy use are two key processes for self-regulatory learning, (Barry J Zimmerman, 1989).

Bjork, Dunlosky, and Kornell (2013) state that learning is increasingly needed outside the formal educational setting and knowing how to manage one's own learning activities is an important survival tool. Unfortunately, our intuitions and introspections are too unreliable to guide us into managing our own learning. They believe that learners need to become "sophisticated" by:

- Understanding key aspects that characterize human learning
- Knowing activities and techniques that help storage and retrieval of information
- Knowing how to monitor one's learning and control one's learning activities to use activities that enhance one's learning

- Understanding biases that can affect judgment of whether learning activities will support later recall and transfer

Investigating Modern Learning Theory in the Computer Programming Classroom

A goal of this research is to formulate a model for teaching and learning computer programming given its peculiar hybrid combination of problem solving and technical skills. The dependent element of interest is the scores students receive on programming exercises while the independent factors include the preferred learning modes of students and different teaching modes as correlated with the student's tendency toward self-regulation.

The proposed method to capture these elements in a research setting proceeds as follows. While the teacher is working through an example programming assignment in class, a recording is made and posted on-line. The students will then have the option of working through the example assignment in the classroom with the instructor or work through the example by watching the video on their own time in a setting and manner of their own choice. It is important to observe that the different choice of venues may be a preference in learning modes or a consequence of their self-regulation tendencies. This research will then model the score earned on the actual assignment as a function of the venue choice with their preference in learning modes, different teaching modes, and their self-regulating nature.

The actual analysis unfolds by first comparing assignment scores within the preferred learning styles to determine if there is a dominant learning style that performs better than other learning styles in programming. Next, a comparison is made between learning modes with preferred venue of working on the example program and self-regulated learning score to see if specific preferred learning modes have an impact on those factors. Also, by modeling assignment scores against the two choices of venue (in-class or on their own) it can be seen if grade increases or decreases. With this information, learning modes can be compared with assignment scores and preferred venue to complete the example program. Also, the correlation between self-regulated learners and preferred learning modes will be analyzed.

Research Questions

To be more specific about the questions under investigation, the following is offered:

1. Given materials in each learning mode, will the students open their preferred learning mode first?
2. Is there a difference in preferred teaching mode (students' ranking) compared to VARK preferred learning modes?
3. Is there a difference in assignment scores that is dependent on preferred learning modes?
4. Is there a difference in assignment scores by venue (in-class or on-line video on their own) when completing the example exercise?
5. Is there a difference in self-regulated learning score by venue (in-class or on-line video on their own) when completing the example exercise?

Summary

Recall that the success of students in a computer programming classroom can be disheartening. One culprit seems to be teachers adopting conventional paradigms in teaching. The premise of this research is that by offering students the materials via all learning modes, their performance will improve/get better. This research will compare students' assignment scores by learning modes when offering teaching materials for each type of learning mode included in VARK®. Hence, computer programming instructors should adopt a teaching pedagogy that includes all modes of teaching available to students; all students would be on a level playing field regardless of their preferred learning mode.

The research will also explore if students designated with a specific preference in learning mode(s), actually use and prefer the same preferred learning mode calculated from taking the VARK® survey. The students are asked at the end of each assignment which teaching mode was most useful to them while completing their assignment and which teaching mode was preferred. If the students choose and prefer to use the same learning mode as their preferred learning mode, hopefully, instructors will offer multiple mode options for their students.

Sharp and Schultz (2013) have suggested that more research is needed to include the use of video and comparing with learning styles. Also, with the recording and posting of the example assignment, students can choose the venue in which they prefer to complete the example, whether it is a preferred learning mode that chooses the venue or they are rated as self-regulated learners and prefer to complete the example on their own due to their self-regulated scores. Comparing assignment scores by venue will ensure that students choosing this method do not do worse on their assignments than those who attend class. Informally, many seasoned teachers have noticed that some students tend to forget the relative importance of how or why programs are written in a certain way when working the example program. If the students use the video while working on their assignment, it will just be another tool for them to succeed.

DEFINITIONS

- **Learning modes** and learning styles are used interchangeably. They describe the format which learners prefer to take in information when learning.
- **Memory (related to programming)** is where data/information is stored/remembered temporarily in a computer program as it is running. Everything stored in memory is lost once the computer program stops running.
- **MSLQ** is a self-reported motivated strategies and learning questionnaire. It can be used to rate college students' motivation and learning strategies for a college course.
- **Pointers (related to programming)** are values that refer to another value stored in memory of the computer program using its memory address.
- **Self-regulated learners** are defined by Barry J Zimmerman (1989) as students that are motivationally and behaviorally active in their own learning process. These students also personally initiate, direct their own efforts, and decisively regulate and select information in various forms to acquire knowledge.
- **Teaching modes and teaching styles** are used interchangeably. They describe the format in which teaching materials are presented to the students.

- **VARK®** is a self-reported, short, easy questionnaire which proposes that it can be used to evaluate preferred learning modes. It has been well-accepted because it is easily understood and its applications are practical. The acronym VARK® stands for the different modes: visual, aural, read/write, and kinesthetic. Visual learners prefer flow charts, graphs, charts, diagrams, and similar types of learning materials. Aural learners prefer discussions, stories, guest speakers, and audio materials. Read/write learners prefer lists, notes, and text in all formats whether printed or on-line. Lastly, kinesthetic learners prefer senses, practical exercises, examples, cases and trial and error learning materials (Neil D Fleming, 2011).

Chapter Two - Literature Review

In this research, the course will present materials of different teaching modes: visual, aural, read/write, and kinesthetic. Students are asked to complete the VARK® survey to evaluate their preferred learning modes. Kinesthetic learning mode is closely related to experiential learning, as they both include learning from experience (Neil D Fleming, 2011; D. A. Kolb, 1984) . Therefore, this section will review previous research dealing with different teaching modes and learning modes. While completing the kinesthetic portion of the different teaching modes, students are given the choice of completing it in class or on-line. This study will examine if preferred learning style or self-regulation can determine which venue the students use to complete the kinesthetic portion. So, previous research on self-regulation will be included in this section.

One of the important parts of this study is working through an example assignment with the students. This type of instruction uses Experiential Learning Theory, which follows next.

Teaching/Learning Modes

Students learn differently. For instance, some students learn better by seeing visual cues while others learn better with verbal cues (Felder, 2010). Felder describes a particular set of these tendencies as a learning mode. Hawk and Shah (2007) discuss that during the last three decades, more attention has been directed toward learning modes. They reviewed and analyzed six different instruments (see Figure 1) that can be used in relation to learning models.

Figure 1: Six Prominent Learning Styles

Note: According to Hawk and Shah (2007)

Hawk and Shah (2007) later summarized each of these in a table (see

Table 1) with a brief description of each learning style for which the model was designed.

Table 1. Learning Model descriptions

Learning Model	Summary of Learning Definitions
Dunn and Dunn Model	How individuals begin to concentrate on, process, internalize, and retain new and difficult information.
Felder and Silverman Learning Style Model	Characteristic strengths and preferences in the ways individuals take in and process information.
Gregorc Learning Style Model	Distinctive and observable behaviors that provide clues about the mediation abilities of individuals and how their minds relate to the world and, therefore, how they learn.
Kolb Experiential Learning	Generalized differences in learning orientation based on the degree to which people emphasize the four modes (Concrete Experience, Reflective Observation, Abstract Conceptualization, and Active Experimentation) of the learning process.
RASI Model	Composite of characteristic cognitive, affective, and psychological factors that serves as an indicator of how an individual interacts with and responds to the learning environment.
VARK® Model	An individual's characteristics and preferred ways of gathering, organizing, and thinking about information. VARK® is in the category of instructional preference because it deals with perceptual modes. Focused on the different ways that we take in and give out information.

Dunn and Dunn Model

The Dunn and Dunn Model defines the learning style by how an individual begins to concentrate on, process, internalize, and retain new and “difficult” information. This model consists of five learning style stimuli with several elements inside each stimulus. The five main stimuli include environmental, emotional, sociological, physiological, and psychological processing. To determine an individual's learning style, a Productivity Environmental Preference Survey (PEPS) is completed. This questionnaire consists of 100 questions covering all five stimuli (Hawk & Shah, 2007).

Felder and Silverman Learning Style

Originating in engineering sciences, it defines learning style as the characteristic strengths and preferences about how people take in and process information. Preferences are among the five bipolar continua: active to reflective, sensing to intuitive, visual to verbal, and sequential to global. The questionnaire used for this learning style is the Index of Learning Styles (ILS). It consists of 44 items that ask participants to choose one of the two endings to a sentence that focuses on some aspect of learning. The questionnaire can be self-administered, self-scored, and self-interpreted (Hawk & Shah, 2007).

Gregorc Learning Style Model

This model works on the idea that people have natural predispositions for learning styles along four bipolar, continuous mind qualities. The Gregorc Style Delineator (GSD) is used to rate people on a score from 10 to 40 in four learning styles: Concrete-Sequential, Abstract-Sequential, Abstract-Random, and Concrete-Random. The individual is asked to put ten sets of four words, corresponding to the four poles just listed, in a specific order. There is, however, limited research on its validity and reliability (Hawk & Shah, 2007).

Kolb Experiential Learning Model

The Kolb Model looks at learning as a holistic set of processes that are continuous and less of an emphasis on outcomes. It consists of four modes of the learning process: Concrete Experience, Reflective Observation, Abstract Conceptualizations, and Active Experimentation. The Kolb Learning Style Inventory (LSI) consists of 12 items. Each item consists of four sentence endings that are ranked. Each of these sentence endings corresponds to one of the four modes in the learning process (Hawk & Shah, 2007).

Buch and Sena (2001), used the Kolb Experiential Learning Model by creating four HTML pages with their content. Each page was designed for each of Kolb's four styles. Their students completed and self-scored the LSI. The researcher assigned the student to one of the four LSI modes. Half of the students were assigned to use the HTML page that matched their LSI mode and the other half assigned a page that did not match their LSI mode. After completing the assignment, the students were given a post-test, measuring their perception of how much the lesson helped them learn and the enjoyment of the design of the teaching mode. In this research, using a MANOVA revealed no significant differences between lesson enjoyment or perceived learning.

Revised Approaches to Studying Inventory (RASI) Model

This model defines learning style as “the composite of characteristic, cognitive, affective, and psychological factors that serves as an indicator of how an individual interacts with and responds to the learning environment” (Hawk & Shah, 2007). Three approaches to studying are involved: deep, surface, and strategic. Scaled measures for these approaches are presented to the individual. The questionnaire consists of 30 or 44 questions where individuals respond with one of five ratings from strongly agree to strongly disagree (Hawk & Shah, 2007).

VARCK® and Felder-Silverman are the only two instruments that are web-based and can easily be administered to students in class or have students complete on their own time and report results. Of these two, VARCK® is the only one that has a moderate support of validity and reliability (Hawk & Shah, 2007). Since the research investigation at hand is going to have students complete and report their results in class and compare teaching and learning modes as it is related to the classroom and VARCK® is in the “category of instructional preference”, this research will categorize students’ preferred learning modes as visual, aural, read/write, and kinesthetic. Next, we will look at previous research that has used the VARCK® questionnaire (Hawk & Shah, 2007).

Visual, Aural, Read-Write, Kinesthetic (VARCK®) Learning Model

An individual’s characteristics and preferred ways of gathering, organizing, and thinking about information VARCK® is in the category of instructional preference because it deals with perceptual modes and is focused on the different ways that we take in and give out information.

Visual learners prefer to learn by pictorial type information and enhancements. This can include graphs, charts, diagrams, and symbols. Students that prefer aural style can “learn by ear” most easily. Aural is the most common mode used by instructors, that is, speaking. Next, read-write are students that prefer to learn by printed word. Lastly, kinesthetic learners like to use all their senses: sight, touch, hear, smell, and taste. These learners also do best by doing (Neil D Fleming, 1995).

Measuring Preferred Learning Modes - VARK®

Neil D Fleming and Mills (1992) completed a study, and after questioning students, found that many students “felt” their learning difficulties were because of how the course material was presented. After these results, they decided to focus on sensory modality as a learning style. To create a technique to focus students’ attention on how they interact with information, they decided that a questionnaire was best. They created the questionnaire to include students’ observations, observations of themselves, and preferences reported by friends. To answer the questions on the questionnaire, the students were asked to pay attention to how they completed a task and after completing the task, all students shared with each other their personal thoughts of how they completed their own tasks and how they saw the other students completing their tasks. At the end of their study, they found that the questionnaire provided a basis for students to reflect on their modal preferences and use information from this reflection to help them adjust their study methods so they are more successful in their studies. This research formed the basis for and the start of the use of visual, aural, reading/writing, and kinesthetic modes (Neil D Fleming, 2011) as a guide to learning styles and concepts.

The VARK® (Visual, Aural, Read/write, or Kinesthetic) questionnaire (<http://vark-learn.com/the-vark-questionnaire/>) is a tool available to researchers to conduct such research. It is a web-based evaluation of how a person completes tasks. It contains 16 multiple choice questions, with four possible answers for each question. Instructions tell the person taking the questionnaire to choose more than one answer if it applies and to skip the question if none of the answers apply. An example of such a question is: “You are planning a vacation for a group. You want some feedback from them about the plan. You would: 1) Phone, text, or email them. 2) Give them a copy of the printed itinerary. 3) Describe some of the highlights they will experience. 4) Use a map to show them the places. The website also warns that the results from the survey should not be rigidly applied but be used to start a discussion (Neil D Fleming, 2016).

Several researchers have used the VARK® questionnaire (Neil D Fleming, 2011) in different areas of study, such as medical students (Hawthorne, Prout, Kinnersley, & Houston, 2009; Lujan & DiCarlo, 2006); surgical students (Kim, Gilbert, & Ristig, 2015); economics students (Leung, McGregor, Sabiston, and Vriliotis (2014); pharmaceutical students (Urval et al. (2014); and English as first or second language students (Reid, 1987). The VARK® questionnaire is completed to evaluate how a person learns best: visual, aural, read/write, or kinesthetic.

To date, research using the VARK® questionnaire (Neil D Fleming, 2011) does not include computer programming students. In addition, the research above only includes one or two different types of teaching modes, not covering all the modes included in VARK®. Sharp and Schultz (2013), stated that some research included video as a teaching mode but not for computer programming.

Since the initial work of Neil D Fleming and Mills (1992), there has been a flurry of research articles about preferred learning styles or modes. Learning styles and modes have been used interchangeably in many research articles (Neil D Fleming, 1995; Neil D Fleming & Baume, 2006; Neil D Fleming & Mills, 1992; Kim et al., 2015; Leung et al., 2014; Lujan & DiCarlo, 2006; Urval et al., 2014). From here out, the term modes will be used in the sense of preferred learning modes.

Neil D Fleming (1995) encourages that when this survey is used with students, it can help instructors to tailor their class materials toward individuals in the class. If instructors complete the survey, they can be motivated to use other modes in the classroom besides their preferred teaching mode. Even if it is impossible to teach each student, it is also impractical to think that a one-size-fits-all teaching method will be good for all students in the class (Felder & Brent, 2005). Tanner and Allen (2004) also suggest that instructors should consider their teaching mode. If instructors acknowledge the existence of different learning modes, they should aspire to use multiple pedagogical styles rather than the teaching mode that worked well for the instructors as learners. Instructors may have different preferred learning modes, therefore, the teaching mode that worked well for them may not be the best teaching mode for their students.

In one study, students took the VARK® survey to find their preferred sensory learning mode. These students were also asked their self-perceived learning modes. The students were made aware of their preferred learning modes, which sometimes were not the same as their self-perceived modes (Urval et al., 2014). Most students were multi-modal in their preferred learning modes (Lujan & DiCarlo, 2006; Urval et al., 2014) which is good news for students and teachers alike. If a student has more than one preferred learning mode, it is more likely that the instructor will use an instructional tactic that aligns with a mode that is helpful for the student.

On the other side of learning, a student knowing their preferred method of learning can adjust the way they study to use their preferred method (Urval et al., 2014). Tanner and Allen (2004) state that it may not be possible or desirable to offer multiple teaching modes, but if students are aware of their learning preferences, they can work toward translating the material used in the classroom to their learning preference.

Lujan and DiCarlo (2006) found that most of their students that completed the VARK® questionnaire preferred more than one mode of learning with only 36.1% preferring just one mode (see

Figure 2). Two of the most used methods in learning are lecturing (auditory) and reading textbooks (reading/writing). Very few students have a preferred learning mode of these two types. Of the 36.1% that had a single mode of learning, only 5% of the students preferred aural learning and 8% preferred reading/writing. The largest group of single mode of learning was kinesthetic, at 18%. These students prefer to use all their senses when learning.

Figure 2. Students by preferred learning mode

There are mixed findings in research about the correlation between learning modes and academic performance. Urval et al. (2014) found no correlation between learning mode preference and academic performance in the medical students involved in their study. This study compared learning styles with self-reported scores on specific tests but does not consider if altering teaching styles affects academic performance.

Kim et al. (2015) found that there was an association with the American Board of Surgery In-Training Examination (ABSITE) scores and learning mode preferences. Students with a read/write preferred learning mode scored higher on the exam than their counterparts with different learning modes. The exam contained only words, which favors students with reading/writing preferred learning modes.

Leung et al. (2014), determined students with a strong preference for the kinesthetic learning mode were the only students that had a significant statistical relationship with final grade. Comparison between preferred learning styles and total percentage grade in the Principles of Macroeconomics course resulted in a positive significant relationship with students' total percentage grade by about three percent.

Sankey, Birch, and Gardiner (2011) conducted an experiment to determine if a variety of teaching modes of material would increase the students' learning outcome within the different learning modes (according to their results from the VARK® questionnaire). After the assignment was completed, students were asked via survey which learning material was the easiest and which material was the most enjoyable. They used six different conditions, each with varying materials of instruction and teaching modes. Sixty students were divided into six groups of ten students. Each group participated in two different conditions. They did not find any significant learning difference across the six groups and six different conditions.

Practicality of Teaching and Learning Modes

Several researchers have discussed learning modes and teaching modes and whether these are important (Felder, 2010; Neil D Fleming, 1995; Neil D Fleming & Baume, 2006; Kirschner, 2017; Urval et al., 2014). Felder (2010) and Hawthorne et al. (2009) argue that including all teaching modes in a course would be impossible and too expensive. Other researchers have concentrated on one teaching style of experiential or kinesthetic but no other learning styles (Alfieri, Brooks, Aldrich, & Tenenbaum, 2011; Dou & Shi, 2015; D. A. Kolb, 1984) which fits in the kinesthetic learning style (Neil D

Fleming, 1995) the most widely rated mode from September – November, 2015 on the VARK® survey website (Neil D Fleming, 2016).

Basic Computer Programming Skills

Researchers have explored what students in computer science majors should learn in their first-year computing course and have compiled a framework of five steps:

1. Abstract the problem from its description. Given a worded problem, the students should be able to create a solution to the problem programmatically.
2. Generate sub-problems. Although the importance/scope of this step is dependent on the design used, a structured program often requires breaking the problem to be solved into several smaller problems.
3. Transform sub-problems into sub-solutions. Students should be able to create error free and correct solutions for each of the sub-problems.
4. Re-compose. The students should be able to take all the sub-solutions and put them together to have a working programming project.
5. Evaluate and iterate. The students should be able to determine if the solution is a good solution, and if not, revisit the coding and improve the solution (McCracken et al., 2001).

A Degree of Closeness was assigned to each computer program, a five-point scale that rated how close the student's computer program was to a working solution. The average degree of closeness was 2.3 out of five points, which is low. This study was conducted for Computer Science majors at several universities (McCracken et al., 2001). Some minimal grasp of programming concepts and skill are required before a student will have the problem-solving skills to accomplish the five-step definition by McCracken et al. (2001) (Lister et al., 2004). Jonassen (2004) states that successful problem-solvers must actively manipulate and test their solutions after they have successfully tested their models in their mind. If Computer Science majors perform problem-solving poorly, what about Business, and other, majors who are required to take a programming course. What can be done to help these students learn more of the basic programming skills and learn how to figure out which constructs to use?

Barriers Programmers Encounter

Ko, Myers, and Aung (2004) observed beginning programmers for a semester and found six main barriers that programmers encounter.

1. *Design barrier*, where the programmer is not sure how to make the program do what it is supposed to do.
2. *Selection barrier*, where the programmer knows what the computer should do but is not sure which programming interface to use.
3. *Coordination barrier*, where the programmer knows which interfaces to use but has trouble getting them to work together.

4. *Understanding barrier*, which are properties of the program's external behavior that conceals what the program did or did not do at compile or run-time, including compilation and run-time errors.
5. *Use barrier*, which refers to the set of properties of a programming interface that hides the way it can be used, how to use it, and what effect such uses will have.
6. *Information barrier*, which occurs when a programmer has a hypothesis about their program's behavior but is not able to test it.

Instructors of programming courses need to find a pedagogic approach that can help students combat these barriers so that they can be successful in the course.

Content to Combat Problems

The content to teach students and the approach of instruction is important.

Reformation of programming courses should look at four things:

1. Adjust teaching idea and course planning to not only pay attention to the basic conceptions, but also help students build thinking methods related to programming.
2. Heightening students' interest and confidence by using small programs they may find interesting.
3. Training students how to have creative thinking related to programming by the instructor demonstrating a working model.
4. Dynamic interaction between task, instructor, and learners by encouraging the students to discuss what needs to be coded for the program to work properly (Gao, 2011).

Recently, instructors have focused on complementing their lecture/lab programming courses with new pedagogical approaches and tools to enhance student learning with software tools that are available through the World Wide Web. A drawback with these tools is that many of them are too hard for beginning programming students to use and consume extra teaching time for the students to use them effectively (Gomez-Albarran, 2005).

Worked Examples

Students learn best when they are required to be actively engaged (Alfieri et al., 2011; Felder & Brent, 2005). Approaches should include guided tasks that include scaffolding to assist students, tasks requiring students to explain their ideas with timely feedback, and/or providing worked examples to complete the task successfully (Alfieri et al., 2011).

In courses where problem-solving is critical, Sweller, van Merriënboer, and Paas (1998) suggest that an emphasis on goal-free problems, worked examples, and completion problems can all be helpful. These pedagogical examples can help students' short-term memory load and transfer the information to long-term memory. Students can also keep these examples that they have worked in their own "library" which allows them to learn from their own personal experiences (Jonassen, 2004).

Caspersen and Bennedsen (2007) describe a worked example as a description of how to solve a problem, including the steps needed for the solution. They found the theory of worked examples should be the foundation for the instructional design of an introductory programming course. They conducted a study comparing programming videos and if on-line or face-to-face students found the videos helpful. They found that there was not a difference in the use of the videos between the two groups.

As stated in the introduction, programming is almost all word problems (Winslow, 1996). Solutions to story “word” problems require the student to convert textual information into a conceptual model that represents the text. With practice and reflection, schemas can become automated (Jonassen, 2003).

Self-Regulated Learning

Barry J Zimmerman (2002) describes a self-regulated learner as a student who does for themselves in a proactive way, and (Barry J Zimmerman, 1989) initiates/directs their own efforts to learn instead of waiting/depending on instructors. Research on self-regulated learning started in the 1970s and early 1980s (B. J. Zimmerman & Schunk, 2001). Students achieving higher grades were more likely to use self-regulatory strategies than those achieving lower grades in a study observation of seventh graders (Pintrich & De Groot, 1990). Personal processes are not the only measure of self-regulated learning, which is also influenced by environmental and behavioral events. Levels of the three variables can be of varying degrees (Barry J Zimmerman, 1989). Hall (2007) displays a social cognitive model of learning (see Figure 3) of the interaction

between personal influences, behavioral influences, and environmental influences that are discussed in self-regulated learning (Hall, 2007; Barry J Zimmerman, 1989).

Figure 3. Social Cognitive Model of Learning

B. J. Zimmerman (2000) updated the social cognitive model of learning to show triadic forms of self-regulation and strategy use vs. feedback loop (see Figure 4).

Figure 4. Triadic Forms of Self-Regulation

There are three cyclic phases of self-regulation learning: forethought, performance or volitional control, and self-reflection. B. J. Zimmerman (2000) created a table that explains two categories for each phase and subcategories (see Table 2).

Table 2: Phase Structure and Sub Processes of Self-Regulation Cyclical Self-Regulatory Phases

Forethought	Performance/volitional control	Self-reflection
Task analysis <ul style="list-style-type: none"> • Goal setting • Strategic planning 	Self-control <ul style="list-style-type: none"> • Self-instruction • Imagery • Attention focusing • Task strategies 	Self-judgment <ul style="list-style-type: none"> • Self-evaluation • Causal attribution
Self-motivation beliefs <ul style="list-style-type: none"> • Self-efficacy • Outcome expectations • Intrinsic interest/value • Goal orientation 	Self-observation <ul style="list-style-type: none"> • Self-recording • Self-experimentation 	Self-reaction <ul style="list-style-type: none"> • Self-satisfaction/affect • Adaptive-defensive

Barry J Zimmerman (1986) devised a set of 14 categories of strategies to use in determining self-regulated learning. The categories of strategy include: self-evaluation, organizing and transforming, goal-setting and planning, seeking information, keeping records and monitoring, environmental structuring, self-consequences, rehearsing and memorizing, seeking social assistance (peers, teachers, adults), reviewing records (tests, notes, textbooks), and others (statements about learning behavior for other than the student, like teachers or parents) (Purdie, Hattie, & Graham, 1996; Barry J Zimmerman, 1986, 1989).

Boekaerts (1999) proposes that learning styles are crucial to describe the quality of self-regulated learning. For self-regulated learning to occur, students need to perceive that there is a choice between modes of processing the information. Then the student can adapt their learning style to accomplish the task or goal. Rowe and Rafferty (2013) suggest that student-regulated learning can be triggered by creating a discussion board at the beginning of the course, prompting students to discuss their expectations, goals, study strategies, and learning styles they may use to complete expectations from the course. A suggestion to have the students complete a survey, such as VARK®, before posting in the discussion board will give students knowledge about their preferred learning styles. After students post the profiles of their plans for studying, the instructor should prompt the students to compare and discuss differences and similarities.

A self-reported questionnaire, Motivated Strategies for Learning Questionnaire, MSLQ, was created by a team of researchers from the National Center of Research to improve learning. The questionnaire has 81 questions that are subdivided into two sections: motivation, and learning strategies. Each of those sections have questions divided into even more sub-categories. Questions can be used as a whole or individually to suit your need and research area (Pintrich, 1991).

Motivated Strategies for Learning Questionnaire (MSLQ)

To determine self-regulation tendencies, the Motivated Strategies for Learning Questionnaire (MSLQ) can be used (Pintrich, 1991). The MSLQ is a self-report questionnaire that can be used to assess college students' motivational tendencies along with the different learning strategies they use. It consists of two major parts: motivation, and learning strategies. There are 15 different scales that can be used together or singly allowing the questionnaire to be modular to fit the needs of the researcher or instructor (Pintrich, 1991). Development of the questionnaire started informally in 1982 while evaluating the effectiveness of a "Learning to Learn" course taught at the University of Michigan (McKeachie, Pintrich, & Lin, 1985; Pintrich et al., 1987). Formal development started in 1986, when the National Center for Research to Improve Postsecondary Teaching and Learning (NCRIPTAL) was founded. Three collaborating institutions were used to collect major waves of data, including 37 classrooms in 14 different subjects. Questions/statements in the questionnaire were revised based on theoretical and empirical analyses (Pintrich, 1991).

Summary

Learning style pedagogy is considering how students acquire and retain information best. Multiple researchers have investigated whether learning styles are important to educating. These research projects have several different outcomes. Some researchers profess that learning styles are important and instructors should consider these different learning styles while preparing teaching materials for their courses, while others who agree with the learning style pedagogy say it is too difficult to cater to each students' preferred learning style. Other researchers have concluded that there is not a scientific or valid process to determine a preferred learning style, while others yet disagree with the learning style pedagogy altogether.

Self-regulated learning is about how much the students take control of their own learning as opposed to relying solely on the instructor. Some research suggests that self-regulated learning is very dependent on learning style pedagogy, while others concentrate on a person's characteristics, such as self-motivation, self-judgement, self-control, and self-reaction.

Chapter Three -- Methodology

Overview of Chapter Three

The purpose of this research was to gather questionnaire results from students to determine their preferred learning modes and offer additional materials to match those different learning modes by reviewing these learning and teaching modes regarding assignment scores, and which of those teaching modes was used and/or helped with working on the assignment to determine if materials offered in a course should be adjusted to students' preferred learning modes.

Research Questions

1. Given materials in each learning mode, will students open their preferred learning mode first?
2. Is there a difference in preferred teaching mode (students' ranking) compared to VARK® preferred learning modes?
3. Is there a difference in assignment scores that is dependent on preferred learning modes?
4. Is there a difference in assignment scores by venue (traditional classroom environment or online video outside of class) when completing the example exercise?
5. Is there a difference in self-regulated learning score by venue (traditional classroom environment or online video outside of class) when completing the example exercise?

Instructional Environment

The course for this study was an introductory programming course that used Visual C# (C Sharp) and Microsoft Visual Studio 2013 to create the programs. Students in the course were enrolled in technical and non-technical degree programs at a four-year university. An information system is a large set of programs and interfaces that provide timely data for business operations and decision-making requirements. There are many important aspects of information systems common to most commercial implementations. In particular, the course focused on the issues of implementation surrounding the system user interfaces.

The course was an undergraduate 14-week course that is taught each fall and spring semester. There is no prerequisite, and many of the students are freshmen and juniors. During the semester, there were ten assignments worth 50 points each, a mid-term and final programming project worth 100 points each, and mid-term and final debugging portions worth 50 points each that totaled 1,000 points for the semester. All assignments also offered five extra credit points by completing an additional, more difficult task. No book was used in the course; instead, Internet links were posted in Blackboard, the Learning Management System, for each assignment. The syllabus is attached as Appendix A.

Overview of weekly activities

- Monday – 8:00 a.m. lecture with power point slides.

- Wednesday – 8:00 a.m. worked through the example assignment in-class with students. Recorded this activity. Tegrity recording made available at 10:00 a.m. Posted assignment description. Posted all additional materials (see Table 3).
- Friday – 8:00 a.m. Example assignment due. Voluntary lab days with no structured lecture or agenda; students work in-class if necessary for help with actual assignment. Actual assignment due Wednesday at noon.

Assignment descriptions (see Appendix E: Assignments and Rubrics) along with documents were posted for each learning mode included in the VARK® questionnaire.

The students completed a questionnaire at the beginning of the research (see Table 3).

Students were able to review the worked example (Jonassen, 2004) while working on their assignments. This learning material was always posted on Wednesday after completing the in-class option. Students that completed the example in class submitted their example assignment at the end of the class. Students who chose to complete the example on-line were required to complete and submit the example assignment before lab class on Friday. Completion of the example assignment was a requirement because it prepared the students for their actual assignment. Also, the Friday lab classes were designed for the students to work on their assignment and receive immediate, individual assistance from the instructor.

Table 3. Teaching-Learning Mode

Learning Mode	Learning Material Posted to Blackboard
V (Visual)	Flowchart
A (Aural)	Audio (mp3) explaining new concept(s)
R (Read/Write)	Text document explaining new concept(s)
K (Kinesthetic)	Worked through example program completed in class or on-line

Visual

Visual learners like information in the form of graphs, charts, and flow diagrams (Neil D Fleming, 1995). One of the problems facing beginning programming students is determining the correct order of the steps needed to determine the correct solution (Lahtinen et al., 2005). A flowchart of the steps in the correct order can help students with this problem; therefore, a flowchart was posted for visual learners in the learning materials. In Assignment 8, links of teaching materials, the flowchart was listed as the first item after the assignment description; in Assignment 9, it was the second item after the description.

The flowchart, seen in Appendix E, is a diagram of how the assignment should be completed. It shows the steps, what to complete in each step, and what needs to happen in the next steps. In Assignment 8, a student sees the following steps according to the flowchart:

1. Click the generate button
2. Clear result text boxes
3. Generate 30 random letters
4. Display to the form
 - a. Letters in ascending order
 - b. Letters in descending order
 - c. Letters in ascending order without duplicates
 - d. Letters in descending order without duplicates

Observing the flow chart for Assignment 9, in Appendix E, the students see two tasks before coding: enter words in text box and click the translate button. Next, the coding begins with the following steps:

1. Pull text from form
2. Fill array by splitting sentence by spaces
3. Send each word in the array to a method that translates into Pig Latin
4. Return translated words from method back to program
5. Fill text box with translated words

Additionally, students can clearly identify the extra credit steps as the boxes colored in yellow. Flowcharts can help the students with their problem of giving insight into how the problem can be broken down (Caspersen & Bennedsen, 2007). Since the flowchart explains the necessary steps to complete the program successfully, the students simply need to convert each step into a program code.

Aural

Aural learning style modality includes preferences in heard or spoken word. These students learn better by hearing and may even talk through a solution while working through the problem (Neil D Fleming, 1995).

For aural learning style, an MP3 audio file describing the new concepts was posted. The transcripts of the audio file can be seen in Appendix E. The audio file for Assignment 8 was almost 13 minutes (12:59), 12KB, while the Assignment 9 audio file was 10 minutes and 26 seconds, 9 KB. In Assignment 8, the audio file was the third learning document after the assignment description and in Assignment 9, it was the first learning document after the assignment description.

MP3 files were posted because they are popular and easy to use. Additionally, they are compressed, which reduces file size and allows faster downloads and a smaller storage footprint. I created these files by reviewing the lecture slides I used for the lecture and recording them on my computer with a headset that included a microphone. This headset placed the microphone close to my mouth and lowered the amount of background noise, which improved the quality of the audio.

Read/Write

Students with this preferred modality prefer the written word whether they are reading or writing the words (Neil D Fleming, 1995). Read/write mode was covered with a text document, included in Appendix E, about the new concepts. The text documents were bulleted explanations about the new concepts, which also included some sample programming text. In Assignment 8, the read/write document was listed first after the description of the assignment and third in Assignment 9.

Although students have links to online reading materials, I put the same information in a document that only included text. This eliminated any visual information in the read/write category. I reviewed the lecture slides while creating the text documents to ensure all material was included.

Kinesthetics

Students with kinesthetic learning mode prefer multisensory learning, which includes learning by doing and learning through application. In this study, completing an example assignment with the instructor talking through the thought process and why/how

to use the new concepts will meet the learning-by-doing that is preferred with this learning style.

Also, this requirement for each assignment was aimed at addressing the issue that students who are actively engaged and included scaffolding perform better (Alfieri et al., 2011). The instructor talked/walked through a program similar to the assigned program. As the class worked through the program, the instructor constantly asked the students what should be done next to make the program work correctly. This worked example (Reed & Bolstad, 1991) was completed to help students through the thought process of solving the problem.

Summary of Materials

Although additional time was needed to create the extra materials, offering alternative forms of materials provides students with more options and more ways to learn. When the material changes from year to year, the additional materials might be too time consuming. However, in courses like programming that do not change often, the materials can be used for more than one semester and make the extra time for creation less costly.

Participants

There were 59 students in the course used for the study. The students, enrolled in an introductory programming course from various degree programs, were 18 – 40 years old, and were full-time students.

Procedure

IRB-approved informed consent form

The study was explained to the 59 students and they were asked to complete an IRB-approved informed consent form at the beginning of the study. Eight of the students did not sign the consent form, so their data were removed from the study. Thus, data was used for 51 students. The consent form is in Appendix B.

Demographic information

Students were asked to complete a demographic survey (Appendix C) to collect previous programming experience, their age, and whether they see themselves as a self-regulating student. The participants were asked to complete this short survey after they signed the consent form. This demographic survey was administered through Qualtrics (<http://qualtrics.com>), which collected their student user ID and allowed me to correlate their responses to other data collected about them during the study.

VARK® Survey

Students completed the VARK® (Visual, Aural, Read/write, and Kinesthetic) survey to collect their preferred learning modes. The VARK® survey is accessible on a Web page at the following URL: <http://vark-learn.com/the-vark-questionnaire/> (Neil D Fleming, 2011). This survey poses questions about the way people prefer to take in and present information. Participants were asked to complete the survey after completing the demographic survey. A copy of the questions on the survey is attached in Appendix D.

After completing the survey, students were given points upon submitting their results (Appendix J) to Blackboard, the course content management system.

The results from the VARK® survey were used to determine the preferred learning mode. If a high number is assigned to a learning mode, it suggests a student's preference. For this study, the highest ranking of the student's preferred learning mode will be used. If the highest mode is a tie (as in this example), both modes will be used.

Teaching and Learning Mode Choices

Neil D Fleming (1995) stated that a normal approach for tertiary students to learn is to involve a variety of strategies to transfer information from short-term to long-term memory. Therefore, during two weeks of the semester, April 3 – April 16, additional information was posted for students to assist in each learning mode.

A flowchart (V – visual from VARK), an audio file explaining the new concept (A – aural), and a reading assignment (R – reading/writing) with just text were posted prior to Monday of each week. With each new concept being taught, a folder was created in Blackboard under the ‘Assignments’ tab. Each of these different teaching modes was posted in the correct folder for both assignments and the different teaching modes were posted in a different order. To determine the order of the different teaching materials, each strategy was assigned one, two, and three, and then a random number generator (<http://www.psychicscience.org/random.aspx>) was used to determine the order. ‘K’ was not assigned a number since it had to be accessed in a different manner. Using Blackboard as the content management system and Tegrity as the software package to record working through the example assignment, there was an issue retrieving the Tegrity recordings - students had to access Tegrity through a tab on the left (Appendix I) instead of the normal listing of the different teaching modes and materials on the same page. The rationale for this setup was to allow retrieval of data that included who and when the recording was accessed.

The students received points for reviewing each different mode in any order they chose. Each Monday was a lecture with Microsoft PowerPoint slides and each Wednesday involved working the example program, keeping a normal routine for the semester. The students were allowed to work the example program in two ways.

The example program could be completed in-class with the instructor readily available to answer questions and assist with debugging, or it could be completed via video outside of class. While working on the example assignment in class, it was recorded. I would pause the recording while I was helping students in the class to avoid a screen not changing and material not being covered. Students were also given the opportunity to work on the example assignment outside of class by watching the video of the in-class walkthrough of the example assignment. The video was posted by 10:00 a.m. each Wednesday, approximately one hour after the end of the class, and the students were required to submit by 9:00 a.m. Friday morning (the beginning of the next class).

Data were retrieved from Blackboard that included the date and time that the students first started reviewing each teaching mode option. Only the first review of the material was collected even though they may have reviewed the material more than once. Using this data and the results of the VARK® questionnaire, researchers could examine the first research question about whether students would open their preferred teaching mode materials first.

Assignments

Assignment description, rubric, example programs, and program solutions are included in Appendix E. Using the grades from these assignments and the results from the VARK® questionnaire, researchers could calculate results for the third and fourth research questions. For this study, assignments 8 and 9 were used because they were assignments with which students had previously had trouble. I did not use Assignment 10

because it needed two days to complete the example program, whereas the other assignments only took one day.

One of the problems students encountered with Assignment 8 was creating random numbers and converting them to letters. Another problem that they encountered was that it also included LINQ (Language Integrated Query), which is very similar to SQL (Structured Query Language), which increased complexity by adding another “language” (database querying language) inside the C# language that we were using for programming. A problem that students encountered with Assignment 9 was understanding how to access specific characters in a string along with creating methods, passing data into a method, and returning data from a method.

Post-Assignment Survey

After both assignments, the students were given points to complete a post-assignment survey (Appendix F) that collected data about their preference of teaching modes and how they used them. These data helped in calculating results for the second research question. The survey data were collected through Qualtrics (<http://qualtrics.com>) and included their student user ID, which allowed correlation of their responses to other data collected about them during the study.

Additional Contact

Fridays were lab days during the semester and contained no structured lecture or agenda. Students worked on their assignment and received help from the professor or teaching assistant. Students were not required to attend class, but were strongly

encouraged to attend if the assignment was not complete. Attendance was noted on the assignments included in the research.

Students would occasionally stop by the instructor's office during office hours for help with their assignment. For additional help with their programs, an assignment was posted on Blackboard named "Need Help?". Students could post their assignments there and email the professor with questions and problems they were encountering.

The professor would email the student back with hints/clues/guidance on how to complete the portion(s) of the students' problems/concerns. An additional form of assistance for the students was on Mondays from noon until 2:00 p.m. with an organized help session where students could come and go as they pleased, but receive help from the professor and teaching assistants. Those who took part in these different types of assistance were noted.

Statistical Data

Table 4 below includes each research question, what data were collected to answer each question, and the type of analysis used with that data. Chi-square was used with research questions that included categorical data (Chi-square is a test for independence when data is categorical data, data not depicted as numbers, and two or more variables are independent).

Table 4. Research Questions, Data Collection Methods, and Analysis Used

Research Question	Data Collection	Analysis
Given materials in each learning mode, will the students open their preferred learning mode first?	Dates/times student retrieved each teaching style mode	Chi-Squared

Research Question	Data Collection	Analysis
	Preferred learning mode collected from the VARK® Questionnaire	Differences in teaching mode chosen first by preferred learning mode
Is there a difference in preferred teaching mode (students' ranking) compared to preferred learning modes based on VARK® results?	Post assignment questionnaire Preferred learning mode collected from the VARK® Questionnaire	t-test $p \leq 0.05$ will prove significant difference in preferred mode. Students that rank preferred/helpful materials the same as their VARK® preferred mode will have higher assignment scores.
Is there a difference in assignment scores that are dependent on preferred learning modes?	Compare scores on assignments within learning modes Assignment scores Preferred learning mode collected from the VARK® Questionnaire	ANOVA $p \leq 0.05$ will prove significant difference in preferred mode. When teaching all modes are offered, is there a difference in success (assignment scores) by learning mode.
Is there a difference in assignment scores by venue (in-class or on-line video) when completing the example exercise?	Compare assignment scores with venue of learning mode Assignment scores Attendance for working the example in class	t-test $p \leq 0.05$ will prove significant difference in either venue. Whether students worked the example in class or on their own did not make a significant difference in their assignment score.

Research Question	Data Collection	Analysis
Is there a difference in self-regulated learning score by venue (in-class or online video on their own) when completing the example exercise?	<p>Compare assignment scores with venue and self-regulated learning score</p> <p>Assignment scores</p> <p>Marked attendance for working the example in class</p>	<p>t-test</p> <p>$p \leq 0.5$ will prove significant difference in assignment score depending on venue.</p> <p>Whether students worked the example in class or on their own did not make a significant difference depending on their self-regulated learning score.</p>

Research Question One

For the research question, “Given materials in each learning mode, will the students open their preferred learning mode first?”, the data required will be date and time that each teaching material is opened, which can be retrieved from Blackboard, and the student’s preferred learning mode from the VARK® questionnaire, which was collected from the students at the beginning of the study. Chi-square was used to calculate independence between teaching material first opened and preferred learning mode calculated by the VARK® questionnaire.

Research Question Two

For the research question, “Is there a difference in preferred teaching mode (students’ ranking) compared to preferred learning modes based on VARK®results?”, the rankings of preferred teaching mode were retrieved from the Qualtrics post-assignment survey. The results submitted by the student from the VARK® questionnaire about the preferred learning were used as well. Pearson Correlation was used to see if there was a correlation between preferred learning mode and the teaching mode rated as the most preferred.

Research Question Three

To arrive at the answer to the third research question, “Is there a difference in assignment scores as dependent on preferred learning modes?”, assignment scores from each assignment were used along with the preferred learning mode submitted from the VARK® questionnaire. A t-test was used with p scores < 0.01 determining a relationship between assignment scores within the different preferred learning modes from the VARK® questionnaire.

Research Question Four

The next research question, “Is there a difference in assignment scores by venue (in-class or online video on their own) when completing the example exercise?”, will use the assignment scores for each assignment along with the venue that each student chose to complete the example assignment. Again, the t-test was used with p scores < 0.01 showing a relationship between the score on the example assignment and the venue used to complete the example assignment.

Research Question Five

The last research question, “Is there a difference in self-regulated learning score by venue (in-class demonstration or online video on their own) when completing the example exercise?”, uses data retrieved from the Motivated Strategies for Learning Questionnaire (MSLQ) and the venue each student used to complete the example assignment. MSLQ was used to calculate a self-regulation score to estimate how likely a student is motivated to learn on their own.

Levene’s Test for Equality of Variances and t-test for Equality of Means were used to calculate the possibility of a relationship between venue chosen and how self-regulated the student is based on their MSLQ self-regulation score (see **Error! Reference source not found.**).

Assumptions

We assumed that using a guided lecture or video to complete the example program would be kinesthetic (‘K’) learning mode, because kinesthetic is defined as “perceptual preference related to the use of experience and practice (simulated or real)” (Neil D Fleming, 2011). By recording the guided lecture, students were given the option to not only complete the example program in class or on their own, but also review the recorded example while working on their assignment to refresh their memory of why and how we used a specific code.

Students will probably use more than one of the additional teaching modes posted, regardless of their preferred learning mode.

Pilot Study

A pilot study was conducted to determine if meaningful data could be collected with the research instruments. The research used a programming course with assignments that were 100 points each, with a total of 1,000 points total for the semester. Data could not be collected from the introductory programming course for the pilot study, so data were instead collected from an advanced programming course. One of the biggest differences between the introductory course and the advanced course was that assignments were worth 50 points and 100 points, respectively. Both have an option to complete a more difficult task for an additional five credit points.

Procedure

The students in the advanced programming course had similar demographics as the introductory course with the exception that they had already taken one programming course. I collected data on one assignment for the pilot study instead of two during the semester for the main research study. Also, for most assignments in the advanced course, the example program that is worked in class with the students takes more than one class period. The assignment used in the pilot study required two days, and I collected data for both days in which we worked on the example program. The pilot study results are displayed below.

Qualitative Comments from Post Survey

Students commented that they liked the ability to complete the in-class assignment on their own time (and would take less time). It took two classroom days (100 minutes) to complete the same assignment that only took 50 minutes outside of class. They also commented that they liked being able to go back in the video if they missed something.

Lessons Learned

There was a problem with the link from the demographic survey to the VARK® questionnaire. Most students missed it, so I had to send it via email. A possible solution to avoid this problem is to post the links. Also, there was a problem with collecting video information. Instead of having the students go directly into Tegrity to view the videos, I created a link to the videos in Blackboard with all the other material. When students used these links, the Tegrity data recorded that the viewer was unknown. I could use Blackboard reports to determine which students clicked on the links and how many times the links were clicked.

The low number of participants could skew the results. Either including courses with more students and/or include more semesters may make a difference in the results. Also, there could be an anomaly because data were collected on only one assignment; including data for more assignments may change results.

Other factors not considered that could influence the results are the amount of time spent on videos, which cannot easily be tracked, and students' motivation and/or anxiety levels. Time spent watching videos can be tracked, but it does not mean the student was actually watching the video the entire time it was running.

Summary of Pilot Study Results

In the pilot study, it was found that most students preferred kinesthetic learning mode, which is not the most widely used teaching mode (Neil D Fleming, 1995), and the least preferred mode is reading/writing, which is the traditional teaching mode by having students read textbooks and give written tests. There was not a significant correlation between preferred learning modes and performance as found previously (Urval et al., 2014). Most likely, the low number of students affected the significance. Students with kinesthetic preferred learning mode were more likely to view the video of the example assignment than attend class. Assignment score average was higher for the students that did not attend class to work on the example assignment compared to those that worked the example assignment in class.

The methodology was revised to more accurately reflect the different teaching modes, and research questions were revised. By adding a specific additional task for each learning mode, along with additional questions in the post-assignment questionnaire, students' preference for teaching mode can be compared with the VARK® preferred learning modes to find similarities and differences.

The pilot study did not consider the possibility that the students may choose to work on the example program independently because they are more inclined to be self-regulated learners. Therefore, during the complete study, this factor was taken into consideration. The students will complete a self-assessment using the MSLQ questionnaire to determine this trait.

Chapter Four – Results

Summary of Chapter Four

The collected data included 59 students enrolled in the course. Of these, 51 students signed consent forms. Two assignments, eight and nine, were used to collect data. One student did not turn in either of those assignments; therefore, the final collected data were from 50 students in the course. To reduce duplicate counts of students that are multi-modal, only the mode with the highest score will be included in the data that concerns each mode. This further reduced multiple counts per student since most students are multi-modal. Additionally, there were nine students who were bi-modal (two modes tied with the same high score) and one student tri-modal (three modes tied with same high score). For students with more than one mode with the same highest score, those students were included for each mode, which will adjust totals to be higher than the total number of students.

Purpose of Study

The purpose of this study was to investigate preferred learning modes of students and choices of teaching modes. I attempted to answer the following questions:

1. Given materials in each learning mode, will the students open their preferred learning mode first?
2. Is there a difference in preferred teaching mode (students' ranking) compared to preferred learning modes determined by the VARK® analysis?

3. Is there a difference in assignment scores based on preferred learning modes?
4. Is there a difference in assignment scores by venue (in-class or online video) when completing the example exercise?
5. Is there a difference in self-regulated learning score by venue (in-class or online video outside of class) when completing the example exercise?

Demographic Data

Students in the course were between the ages of 18 and 35, with the largest segment between the ages of 18 and 21.

Figure 5: Ages of Participants

The course also consisted of more males (32) than females (18).

Figure 6: Gender of Participants

Learning Modes

Figure 7 shows learning modes of the students after taking the VARK® survey (Neil D Fleming, 2011). Most students (68%) are multi-modal, which means they have more than one preferred learning mode. However, there are a few (32%) who are uni-modal, having only one preferred learning mode. Very few students (4%) have only a read/write preferred learning mode, which has been the traditional type of instruction, requiring students to read textbooks. The largest preferred learning mode is kinesthetic, uni-modal or multi-modal (20% and 59% respectively). Kinesthetic has been the least-used teaching mode (Neil D Fleming, 1995).

Figure 7: Population by Preferred Learning Mode (VARK® survey)

Of the 49 students that completed the demographic survey and consented to the study, four females were bi-modal (two prominent preferred learning modes), five males were bi-modal, and one female was tri-modal (three prominent preferred learning modes). Therefore, there are 60 data points when comparing gender with preferred learning mode.

When analyzing gender by preferred learning mode from the VARK® questionnaire, there was not a difference in reading/writing (seven of each females and males) and kinesthetic (12 female and 10 male). There were greater differences in visual (three female and nine male) and aural (one female and 11 males).

Table 5: Gender of Participants by Preferred Learning Mode

Mode * Gender Crosstabulation				
		Gender		Total
		F	M	
Mode	V	3	9	12
	A	1	11	12
	R	7	7	14
	K	12	10	22
Total		23	37	60

Using Chi-Square calculations for the categorical data, the Asymptotic Significance is less than .05, suggesting a relationship between gender and preferred learning mode results from the VARK® questionnaire.

Table 6: Chi-Square Tests. Gender by Preferred Learning Mode

Chi-Square Tests			
	Value	df	Asymptotic Significance (2-sided)
Pearson Chi-Square	8.723 ^a	3	.033
Likelihood Ratio	9.776	3	.021
N of Valid Cases	60		

a. 2 cells (25.0%) have expected count less than 5. The minimum expected count is 4.60.

Students' Order of Choosing Teaching Modes

To answer the first research question, I gathered dates and times of when the students accessed each type of teaching mode. Using this data, I could determine if the students' first selected teaching mode was the same as their self-reported preferred learning mode. The order in which the choices are posted could play a part in the order the students chose.

The students could retrieve teaching documents through a list of options that were posted in Blackboard. Therefore, the different options were posted in a different order for each assignment (see Appendix H). The recorded video of the example assignment was stored separately and contained data about which students watched the video, when, and for how long. It was located under the Tegrity tab in Blackboard (see Appendix I).

In this study, most students did not first choose the teaching style that matched their preferred learning mode. Also, of the 50 students, three in each of the two assignments did not view any of the different modes of teaching materials posted. Of the 47 remaining students, only 17 and 20 - for assignments 8 and 9 respectively - first chose the materials for the assignment that matched their preferred learning mode. Nine of the students were bi-modal, two modes tied for most prominent, and one student was tri-modal. Three modes tied for most prominent, VRK. Each mode of the multi-modal students was included in the calculations; therefore, 47 initial associations with VARK® preferred along with 9 more for bi-modal and 2 for tri-modal for a total of 58 data points (see Table 5).

Of the 58 VARK® and first selected modes for Assignment 8, the students who matched their first selected VARK® preferred mode were: one visual, zero auditory, six read/write, and 11 kinesthetic. Both read/write and kinesthetic materials had 28 associations with a VARK® preference.

Table 7: Assignment 8. VARK® Mode vs. First Mode Accessed

**VARK® Mode * First Mode Accessed Cross Tabulation
Assignment 8**

Mode Reviewed First vs. VARK® Mode	First Mode Accessed				Total
	V	A	R	K	
VARK® Preferred Mode V	1	0	5	7	13
A	0	0	5	4	9
R	0	0	6	6	12
K	1	0	12	11	24
Total	2	0	28	28	58

For the categorical data Chi-Square test statistic was applied to the data relating to first accessed. The Asymptotic Significance is greater than .05; thus, there is not a significant relationship between a student's preferred VARK® learning mode and the mode of material the students would access first (See Table 6).

Table 8: Assignment 8. Chi-Square Statistics. VARK® Mode vs. Mode First Accessed

**Chi-Square Tests
Assignment 8**

Mode Accessed First vs. VARK® Preferred Mode	Value	df	Asymptotic Significance (2-sided)
Pearson Chi-Square	1.900	6	.929
Likelihood Ratio	2.453	6	.874
N of Valid Cases	58		

a. 6 cells (50.0%) have expected count less than 5. The minimum expected count is .31.

Of the 58 data points in Assignment 9, the number of students that first accessed materials that matched their VARK® preferred mode were zero visual, zero auditory, three read/write, and 17 kinesthetic. Kinesthetic materials had 42 associations with a VARK preference (see Table 7).

Table 9: Assignment 9. VARK® Mode vs. First Mode Accessed

VARK® Mode * First Mode Accessed Cross Tabulation

Assignment 9

Mode Reviewed First vs. VARK® Mode	First Mode Accessed				Total
	V	A	R	K	
VARK® V	0	1	1	10	12
Preferred Mode A	0	0	3	7	10
R	1	1	3	8	13
K	1	1	4	17	23
Total	2	3	11	42	58

The Chi-Square statistic was applied to the data without finding any relationship between a student's preferred learning mode and the mode of material first accessed. The Asymptotic Significance is above 0.05 (see

Table 10).

Table 10: Assignment 9. Chi-Square Statistics. VARK® Mode vs. Mode First Accessed

Chi-Square Tests

Assignment 9

Mode Reviewed First vs. VARK® Mode	Value	df	Asymptotic Significance (2-sided)
Pearson Chi-Square	4.359	9	.886
Likelihood Ratio	5.487	9	.790
N of Valid Cases	58		

12 cells (75.0%) have expected count less than 5. The minimum expected count is .34.

Students' Rank of Teaching Modes Compared with Preferred Learning Mode

In the post-assignment survey, students were asked to rank the different teaching modes in order of helpfulness on the assignment, and their preferred teaching mode. The second research question compares the students' ranking of teaching modes with their preferred learning mode. Twenty-nine students completed the post-assignment survey for Assignment 8, and 32 students completed the post-assignment survey for Assignment 9. Kinesthetic was ranked most helpful and preferred in both assignments. In Assignment 8 with 22 students ranking kinesthetic as most helpful and preferred, while in Assignment 9, 21 students ranked kinesthetic as most helpful and 20 students ranked kinesthetic as preferred.

Verbal and aural were both ranked the lowest for most helpful and preferred on Assignment 8, with only two students ranking each as the highest. These were closely followed by reading/writing, with three students ranking both most helpful and preferred.

On Assignment 9, aural and reading/writing were ranked the lowest as most helpful and preferred. Three students ranked aural as most helpful and two students ranked aural as preferred. Zero students ranked read/write as most helpful and two students ranked read/write as preferred teaching material. Visual was in the middle with eight for most helpful and preferred (see Table 11).

Table 11: Summary of Most Helpful and Preferred Rankings by Students

Mode	Assignment 8		Assignment 9	
	Most Helpful	Preferred	Most Helpful	Preferred
V	2	2	8	8
A	2	2	3	2
R	3	3	0	2
K	22	22	21	20
Totals	29	29	32	32

Comparing students' rankings with their preferred learning modes from the VARK® questionnaire, Assignment 8 rankings were closer to the preferred learning mode. Most helpful rankings matched preferred learning mode 13 out of 29 times, and most preferred rankings matched preferred learning mode 14 out of 29 times.

In Assignment 9, there was a bigger difference between rankings and preferred teaching/learning mode results from the VARK® questionnaire (see Table 12). Ten most helpful rankings matched and 11 most preferred rankings matched preferred learning mode results of 32 observations.

Table 12: Student Ratings of Teaching Modes Material Compared to Preferred Learning Modes

Ratings that Match Preferred Learning Mode				
	Assignment 8		Assignment 9	
	Most Helpful Matches	Most Preferred Matches	Most Helpful Matches	Most Preferred Matches
Matched	13	14	10	11
Did Not Match	16	15	22	21
Total	29	29	32	32

The next tables show the scores students earned on the assignments compared by whether their rankings matched their preferred learning mode from the VARK® questionnaire. In other words, did the students who ranked the material most preferred the same as their preferred teaching/learning mode, score better on their assignment than those who ranked different materials than their preferred learning mode materials?

In Assignment 8, there were 14 students that matched by ranking the most preferred materials the same mode as their learning mode from the VARK® questionnaire and 15 students that did not match the preferred materials the same as their learning mode from the VARK® questionnaire. Comparing assignment scores for students that chose the same preferred teaching mode as their preferred learning mode to those that did not match, the mean between the two groups were close with a 48.2857 for students that matched and 50.2 for students that did not match (see Table 11).

Table 13: Group Statistics. Assignment 8 Scores. Mode Ranked Most Preferred vs. VARK® Preferred Mode

Assignment 8 Scores Ranked Most Preferred Group Statistics					
	Matched/Unmatched	N	Mean	Std. Deviation	Std. Error Mean
Score	Matched	14	48.2857	9.98020	2.66732
	Unmatched	15	50.2000	6.87854	1.77603

In Assignment 8, there were 13 students that matched by ranking the most helpful materials the same mode as their learning mode from the VARK® questionnaire and 16 students that did not match the preferred materials the same as their learning mode from the VARK® questionnaire. The mean of assignment scores between the two groups was close with a 47.7692 for students that matched and 50.5 for students that did not match (see Table 12).

Table 14: Group Statistics. Assignment 8 Scores. Mode Ranked Most Helpful vs. VARK® Preferred Mode

Assignment 8 Scores Ranked Most Helpful Group Statistics					
	Matched\Unmatched	N	Mean	Std. Deviation	Std. Error Mean
Scores	Matched	13	47.7692	10.19112	2.82651
	Unmatched	16	50.5000	6.75278	1.68819

By implementing a t-test for Assignment 8 scores of students ranking of the preferred materials matching or not matching their preferred learning mode, we can determine if there is a relationship or not between assignment scores of students that chose materials preferred as matching or not matching the students' preferred learning mode (see

Table 15). If the value in the Sig. column of Levene's Test for Equality of Variances is greater than .05, we look at the top row of the rest of the table to see if there is a statistically significant difference. Observation of the top number in the Sig. (2-tailed) column, the number .55, which is greater than .05, determines that there is not a statistically significant difference between the assignment scores and the differences are due to chance. Therefore, the difference in assignment scores is only by chance and not determined whether they ranked the preferred materials that were their preferred learning mode.

Table 15: T-Test: Assignment 8 Scores. Mode Ranked Preferred vs. VARK® Preferred Mode

Assignment 8 Scores									
Ranked Most Preferred									
Independent Samples Test									
Levene's Test for Equality of Variances		t-test for Equality of Means							
Score	Equal variances assumed	F	Sig.	t	df	)	Difference Mean	Std. Error Difference	95% Confidence Interval of the Difference
	Equal variances not assumed								
		1.952	.174	-.605	27	.550	-1.91429	3.16396	-8.40620 4.57763
				-.597	22.902	.556	-1.91429	3.20451	-8.54488 4.71631

The t-test for Assignment 8 of students' ranking of the most helpful materials matching or not matching their preferred learning mode can determine if there is a relationship between assignment scores of students that chose materials most helpful as matching or not matching the student's preferred learning mode (see Table 16). If the value in the Sig. column in Levene's Test for Equality of Variances is greater than .05, then we look at the top row of the rest of the table to see if there is a statistically significant difference. Observation of the top number in the Sig. (2-tailed) column, the number .395, which is greater than .05 determines that there is not a statistically significant difference between the assignment scores and the differences are due to chance. Therefore, the difference in assignment scores is only by chance and not determined by whether they ranked the most helpful materials that matched their preferred learning mode.

Table 16: T-Test: Assignment 8 Scores. Mode Ranked Most Helpful vs. VARK® Preferred Mode

Assignment 8 Scores Ranked Most Helpful Independent Samples Test									
		Levene's Test for Equality of Variances		t-test for Equality of Means					
		F	Sig.	t	df	(2- tailed	Mean Difference	Std. Error Diff.	95% Confidence Interval of the Difference
Scores	Equal variances assumed	2.823	.104	-.865	27	.395	-2.73077	3.15718	-9.20876 3.74722
	Equal variances not assumed			-.829	20.048	.417	-2.73077	3.29229	-9.59731 4.13577

In Assignment 9, there were 11 students that matched by ranking the most preferred materials the same mode as their learning mode from the VARK® questionnaire and 21 students that did not match the preferred materials the same as their learning mode from the VARK® questionnaire. The mean, of the assignment score, between the two groups was close with a 50.7727 for students that matched their preferred ranked teaching mode with their preferred learning mode and 52.4286 for students that did not match (see Table 17).

Table 17: Group Statistics. Assignment 9 Scores. Mode Ranked Preferred vs. VARK® Preferred Mode

Assignment 9 Scores Ranked Preferred Group Statistics					
	Matched/Unmatched	N	Mean	Std. Deviation	Std. Error Mean
Scores	Matched	11	50.7727	8.81296	2.65721
	Unmatched	21	52.4286	3.05486	.66662

In Assignment 9, there were 10 students that matched by ranking the most helpful materials the same mode as their learning mode from the VARK® questionnaire and 22 students that did not match the preferred materials the same as their learning mode from the VARK® questionnaire. The mean between the two groups was close with a 52.35 for students that matched and 51.6364 for students that did not match (see Table 18).

Table 18: Group Statistics. Assignment 9 Scores. Mode Ranked Most Helpful vs. VARK® Preferred Mode

**Assignment 9 Scores
Ranked Most Helpful
Group Statistics**

	Matched/Unmatched	N	Mean	Std. Deviation	Std. Error Mean
Scores	Matched	10	52.35	3.38337	1.06992
	Unmatched	22	51.6364	6.46084	1.37746

Observation of the t-test for Assignment 9 scores of students ranking of the preferred materials matching or not matching their preferred learning mode can suggest whether there is a relationship between assignment scores of students that ranked materials preferred as matching or not matching the student's preferred learning mode (see Table 19). Since the value in the Sig. column under Levene's Test for Equality of Variances is greater than .05, we look at the top row of the rest of the table to see if there is a statistically significant difference. Observation of the top number in the Sig. (2-tailed) column, the number .439, which is greater than .05 determines that there is not a statistically significant difference between the assignment scores and the differences are due to chance. Therefore, the difference in assignment scores is only by chance and not determined by whether they ranked the preferred materials that matched their preferred learning mode.

Table 19: T-Test: Assignment 9 Scores. Mode Ranked Preferred vs. VARK® Preferred Mode

Assignment 9 Scores Ranked Preferred Independent Samples Test										
Levene's Test for Equality of Variances					t-test for Equality of Means					
					Sig. (2- tailed	Mean	Std. Error	95% Confidence Interval of the Difference		
F	Sig.	t	df	)	Difference	Difference	Lower	Upper		
Scores	Equal									
	variances	2.486	.125	-7.785	30	.439	-1.65584	2.10909	-5.96318	2.65149
	assumed									
	Equal									
	variances									
	not									
	assumed									

Observation of the t-test for Assignment 9 scores of students' ranking of the most helpful materials matching or not matching their preferred learning mode can suggest if there is a relationship or not between assignment scores of students that ranked materials most helpful as matching or not matching the students' preferred learning modes (see Table 20). If the value in the Sig. column under Levene's Test for Equality of Variances is greater than .05, we look at the top row of the rest of the table to see if there is a statistically significant difference. Observing the top number in the Sig. (2-tailed) column, the number .746, which is greater than .05 determines that there is not a statistically significant difference between the assignment scores and the differences are due to chance. Therefore, the difference in assignment scores is only by chance and not determined by whether they ranked the most helpful materials matched their preferred learning mode.

Table 20: T-Test: Assignment 9 Scores. Mode Ranked Most Helpful vs. VARK® Preferred Mode

Assignment 9 Scores Ranked Most Helpful Independent Samples Test									
Levene's Test for Equality of Variances				t-test for Equality of Means					
		Sig. (2- tailed		Mean	Std. Error	95% Confidence Interval of the Difference			
F	Sig.	t	df	)	Difference	Difference	Lower	Upper	
Scores	Equal								
	variances	.331	.570	.327	30	.746	.71364	2.17937	-3.73723 5.16450
	assumed								
	Equal								
	variances								
	not								
	assumed	.409	29.191	.685	.71364	1.74416	-2.85256	4.27984	

Assignment Scores by Preferred Learning Modes

Next, observation of the third research question, determining if the assignment scores would vary depending on the students' preferred learning mode. Assignments 8 and 9 were both worth 50 points, each with an additional 5 possible credit points (for a total of 55 points).

The descriptive analysis for scores for Assignment 8 by preferred learning mode are in the table below. The data consists of 54 scores with eight scores in aural, 12 scores for visual, 13 scores for reading/writing, and 21 in kinesthetic. The 54 scores included 47 students that submitted an assignment. Of those 47 students, three students did not submit a consent form for the study, leaving 44 students' scores for Assignment 8 data. Eight students were multimodal (two preferred learning modes) along with one student that was tri-modal (three preferred learning modes). The assignment score means, by preferred learning style, ranged from 47.3571 to 52.25 (see Table 19).

Table 21. Assignment 8 Scores by Learning Mode

Assignment 8			
Assignment Scores by Learning Mode			
Descriptive Analysis			
Preferred Mode	Mean	N	Std. Deviation
V	50.2083	12	6.87042
A	52.2500	8	2.53546
R	47.7308	13	8.66414
K	47.3571	21	10.50850
Total	48.8056	54	8.52434

The descriptive analysis for scores for Assignment 9 by preferred learning mode (see Table 22). The data consists of 56 scores with 10 scores in aural, 11 scores for visual, 12 scores for reading/writing, and 23 in kinesthetic. The 56 scores include 49 students that submitted the assignment. Of those 49 students, two students did not sign the consent form, leaving 47 students' scores to be included in the study. Seven of these students were bimodal (two preferred learning modes) and one was tri-modal (three preferred learning modes). The assignment score means, by preferred learning modes, ranged from 51.2391 to 53.375.

Table 22. Assignment 9 Scores by Learning Modes

Assignment 9 Scores Assignment Scores by Learning Mode Descriptive Analysis			
Preferred Mode	Mean	N	Std. Deviation
V	53.0000	11	2.19089
A	53.3500	10	2.62520
R	53.3750	12	2.99336
K	51.2391	23	6.38156
Total	52.4196	56	4.59226

The data has more than two variables, thus a t-test could not be used. An ANOVA table was used to calculate if there is a significant difference in the means. Observation of the table, the significance number, or p value, of 0.497 is greater than 0.05; therefore, there is not a difference in the assignment scores depending on their preferred learning mode (see Table 23).

Table 23. ANOVA Table for Assignment 8 Scores

Assignment 8 Scores

ANOVA Table

		Sum of Squares	df	Mean Square	F	Sig.
Scores * Preferred Mode	Between Groups (Combined)	177.600	3	59.200	.806	.497
	Within Groups	3673.608	50	73.472		
	Total	3851.208	53			

There are more than two groups, thus an ANOVA table is appropriate. The significance, or p value, of 0.463 is larger than 0.05, which suggests that the difference in assignment scores are not due to preferred learning mode (see Table 24).

Table 24. ANOVA Table for Assignment 9 Scores

Assignment 9 Scores

ANOVA Table

		Sum of Squares	df	Mean Square	F	Sig.
Scores * Preferred Mode	Between Groups (Combined)	55.366	3	18.455	.869	.463
	Within Groups	1104.522	52	21.241		
	Total	1159.888	55			

Preferred Learning Modes by Choice of Venue for Example Program

When completing the kinesthetic portion of the assignment, students were given the option to complete the scaffolded example in class or on-line by watching the recorded in-class walk-through.

Forty-four students submitted example Assignment 8. Thirty students completed the example assignment in class, while 14 completed the example assignment online. Nine of the students were bi-modal (two preferred modes), and one student tri-modal (three preferred modes). Most of the preferred learning modes that participated in completing the example in class were kinesthetic, 17, while most that completed the example online were reading/writing mode, seven (see Table 25).

Table 25. Example Assignment 8 Data

Example Assignment 8				
Preferred Mode * Venue Crosstabulation				
		Venue		Total
		In class	Online	
Preferred Mode	V	8	4	12
	A	7	2	9
	R	6	7	13
	K	17	4	21
Total		38	17	55

For Assignment 9, 46 students completed the example assignment. Twenty-seven completed the example in class, while 19 completed the example online. Nine of the students were bi-modal (two preferred modes) and one student tri-modal (three preferred modes). Most of the preferred learning modes that participated in completing the example in class were kinesthetic, 17, while most that completed the example online were reading/writing mode, nine (see Table 26).

Table 26. Example Assignment 9 Data

Example 9 Assignment Preferred Mode * Venue Crosstabulation				
		Venue		Total
		In class	Online	
Preferred Mode	V	7	4	11
	A	6	4	10
	R	4	9	13
	K	17	6	23
Total		34	23	57

Chi-Square statistic was used to calculate if there is a statistical difference in mode and venue (see Table 27). If the Pearson Chi-Square p-value Asymptotic Significance is above 0.05 (0.176 for Assignment 8), then no association can be found between preferred learning mode and whether the student attends class for the example assignment. Table 25 shows the Pearson Chi Square calculations comparing students' preferred learning mode with the venue they chose to complete the example assignment. With an asymptotic significance greater than 0.05, the choice of venue is by chance and related to the students' preferred learning mode.

Table 27: Chi-Square. Example Assignment 8. Preferred Mode * Venue

Example Assignment 8
Preferred Mode * Venue
Chi-Square Tests

	Value	df	Asymptotic Significance (2-sided)
Pearson Chi-Square	4.937 ^a	3	.176
Likelihood Ratio	4.815	3	.186
N of Valid Cases	55		

a. 3 cells (37.5%) have expected count less than 5. The minimum expected count is 2.78.

Chi-Squared statistic was used to calculate statistical difference in mode and venue. If the Pearson Chi-Square p-value Asymptotic Significance is above 0.05 (0.089 for Assignment 9), then no association can be found between preferred learning mode and whether the student attends class for the example assignment. Table 28 shows the Pearson Chi-Square calculations and with the asymptotic significance larger than 0.05, the choice of venue is purely by chance and not related to the students' preferred learning style.

Table 28: Chi-Square. Assignment 9. Preferred Mode * Venue

Example Assignment 9 Preferred Mode * Venue Chi-Square Tests			
	Value	df	Asymptotic Significance (2-sided)
Pearson Chi-Square	6.522 ^a	3	.089
Likelihood Ratio	6.551	3	.088
N of Valid Cases	57		

a. 2 cells (25.0%) have expected count less than 5. The minimum expected count is 4.04.

Assignment Scores by Venue Used to Complete Example Assignment

Offering students multiple teaching modes (for them to access outside of class) is helpful to them because most tertiary students use multiple modes of teaching to solidify their learning (Neil D Fleming, 1995). When offering the different modes (outside of class), we do not want the students' scores on assessments to decrease. Therefore, we

want to compare assignment scores of students that attended class to those that completed the example program on their own outside of class.

While working on the Assignment 8 example, 30 students attended class, while 14 completed the example online. The assignment mean of the students that completed the example online was somewhat higher than that of the students that completed the example in class, 50.9643 and 48.0333 respectively (see Table 29).

Table 29. Assignment 8 Statistics. Example Venue * Assignment Score

Assignment 8 Example Venue * Assignment Score Group Statistics					
	Venue	N	Mean	Std. Deviation	Std. Error Mean
Assignment Score	In class	30	48.0333	8.85898	1.61742
	Online	14	50.9643	6.76377	1.80769

Using a t-test, since the Significance column is 0.208 and higher than 0.05, the Significance (2-tailed) in the top row is compared to 0.05. The 0.208 is higher than 0.05, which suggests there is not a relationship between the venue students used for the example and their assignment score. Therefore, students' assignment scores cannot be determined by the venue students use to complete the example assignment (see Table 30).

Table 30. Assignment 8 t-test Calculations. Example Venue * Assignment Score

Assignment 8									
Example Venue * Assignment Scores									
Independent Samples Test									
Levene's Test for Equality of Variances									
t-test for Equality of Means									
		Sig. (2-tailed)		Mean Diff.		Std. Error Diff.		95% Confidence Interval of the Difference	
F	Sig.	t	df					Lower	Upper
S Equal									
c variances	1.636	.208	-1.095	42	.280	-2.93095	2.67591	-8.33115	2.46924
o assumed									
r Equal									
e variances			32.						
not	-1.208	74			.236	-2.93095	2.42566	-7.86748	2.00557
assumed			0						

For Assignment 9, there were 27 students that attended class to work on the example assignment, while 19 chose to complete the example online. Observation of the descriptive statistics below (see Table 29), it can be seen that the means between the groups were very close; there were 52.3333 in-class and 52.1842 online.

Table 31. Assignment 9 Statistics. Example Venue * Assignment Score

<p style="text-align: center;">Assignment 9 Example Venue * Assignment Score Group Statistics</p>					
	Venue	N	Mean	Std. Deviation	Std. Error Mean
Assignment Score	In class	27	52.3333	2.90556	.55918
	Online	19	52.1842	6.78071	1.55560

The significance of 0.919 is higher than 0.05, which suggests that there is not a relationship between the venue students used for the example and their assignment score. Therefore, students' assignment scores cannot be determined by the venue students use to complete the example assignment (see Table 32).

Table 32. Assignment 9 t-test calculations. Example Venue * Assignment Score

Assignment 9

Example Venue * Assignment Scores

Independent Samples Test

	Levene's Test for Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	df	Sig. (2-tailed)	Mean Diff.	Std. Error Diff.	95% Confidence Interval of the Difference	
								Lower	Upper
S Equal c variances o assumed r Equal e variances not assumed	.627	.433	.102	44	.919	.14912	1.46079	-2.79491	3.09316
			.090	22.690	.929	.14912	1.65305	-3.27306	3.57131

Self-Regulated Score by Venue

The last question examines why a student may prefer to work on the example assignment outside of the classroom with online video instead of in the classroom with live instructor.

There was not a difference between learning styles whether the student chose to complete the example assignment in or outside of the classroom. This question examines whether their self-regulation plays a factor in that decision.

The data in this table uses data from Assignment 8. Fourteen students completed the example program online while 30 attended class. The average of the MLSQ score for those completing the example by watching the video was slighter higher than the students that attended class (see Table 33).

Table 33: Assignment 8 Group Statistics. MLSQ Score * Venue of Completing Example

Assignment 8 MLSQ Score * Venue of Completing Example Group Statistics					
	In Class	N	Mean	Std. Deviation	Std. Error Mean
MLSQ	No	14	4.3750	.48893	.13067
	Yes	30	4.0694	.83651	.15273

With this data, an independent sample T-test was calculated. Since the value in the Sig. column, under Levene's Test for Equality of Variances is greater than .05, we look at the top row to see if there is a statistically significant difference. Observation of the top number in the Sig. (2-tailed) column, the number .213, which is greater than 0.05 determines that there is not a statistically significant difference between the numbers and the differences are due to chance and not determined by the students' self-regulation score (see Table 34).

Table 34. Assignment 8 t-test. MLSQ * Venue for Example

Assignment 8 MLSQ Score * Venue of Completing Example Independent Samples Test									
	Levene's Test for Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
								Lower	Upper
MLSQ Equal variances assumed	3.075	.087	1.265	42	.213	.30556	.24160	-.18200	.79312
MLSQ Equal variances not assumed			1.520	39.627	.136	.30556	.20100	-.10079	.71191

Observation of assignment nine, 27 students attended class for the example assignment while 19 completed the example by watching the Tegrity video. The mean MLSQ score was slightly larger with the students that completed the example online versus those that completed the example assignment in class (see Table 35).

Table 35. Assignment 9 Group Statistics. MLSQ * Venue of Completing Example

Assignment 9 MLSQ Score * Venue of Completing Example Group Statistics					
	Venue	N	Mean	Std. Deviation	Std. Error Mean
MLSQ Score	In Class	27	3.9475	.95635	.18405
	Online	19	4.3158	.60001	.13765

With this data, an independent sample T-test was calculated. Observation of the bottom number in the Sig. (2-tailed) column, the number .116, which is greater than 0.05, determines that there is not a statistically significant difference between the numbers and that the differences are due to chance. Therefore, there is not a relationship between self-regulation score and which venue the student will use to complete the example assignment (see Table 36).

Table 36. Assignment 9 t-test. MLSQ * Venue for Example

Assignment 9 MLSQ Score * Venue of Completing Example Independent Samples Test									
Levene's Test for Equality of Variances				t-test for Equality of Means					
							95% Confidence Interval of the Difference		
	F	Sig.	t	df	Sig. (2- tailed)	Mean Difference	Std. Error Differ	Lower	Upper
MLSQEqual Score variances assumed	4.969	.031	-1.483	44	.145	-.36826	.24833	-.86873	.13222
Equal variances not assumed			-1.602	43.543	.116	-.36826	.22983	-.83159	.09507

Working on Assignments

One question on the post-assignment survey asked the students which of the materials posted in Blackboard they referred to while working on the actual assignment. Twenty-four students completed the survey after Assignment 8. Three students did not use any materials while working on their assignment. Seven students used two materials, eleven students used three items, and three students used three of the materials (see Table 37).

Table 37: Assignment 8. Number of Materials Used While Working on Assignment

Number of Materials Used While Working on Assignment 8			
0 Items	1 Item	2 Items	3 Items
3	7	11	3

The flowchart and text files were most widely used while working on the assignment (23 and 22 respectively). Some students noted that they used those files because it was easier to find specific information quickly instead of scanning through the video or audio. The next most used was the walk-through of the example assignment (i.e. Tegrity video), used by nine students, with audio file being used the least (by five students) (see Table 36).

Table 38: Assignment 8. VARK® Mode * Material Used While Working on Assignment

**VAR K® Mode * Material Used While Working on Assignment
Assignment 8**

Material Used While Working on Assignment vs. VARK® Mode	Material Used While Working on Assignment				Total
	Flowchart (V)	Audio File (A)	Text File (R)	Example (K)	
VAR K® V	5	1	9	1	16
Preferred A	4	0	4	1	9
Mode R	3	1	3	2	9
K	11	3	6	5	25
Total	23	5	22	9	59

Forty-two students completed the survey after Assignment 9. Six of these students did not use any of the materials while working on their assignment, 15 used one item, 13 used two items, seven used three items, and one person used all four of the materials to help complete the assignment (see Table 39).

Table 39: Assignment 9. Number of Materials Used While Working on Assignment

**Number of Materials Used While Working on
Assignment 9**

0 Items	1 Item	2 Items	3 Items	4 Items
6	15	13	7	1

The flowchart and text files were most widely used, 26 and 22 respectively, while working on the assignment. Next was the walk-through of the example assignment, used by 11, with audio file being used the least, by seven students (see Table 40).

Table 40: Assignment 9. VARK® Mode * Material Used While Working on Assignment

**VARK® Mode * Material Used While Working on Assignment
Assignment 9**

Material Used While Working on Assignment vs. VARK® Mode	Material Used While Working on Assignment				Total
	Flowchart (V)	Audio File (A)	Text File (R)	Example (K)	
VARK® V	3	2	6	3	14
Preferred A	4	1	6	0	11
Mode R	6	1	5	3	15
K	13	3	5	5	26
Total	26	7	22	11	66

Some students indicated that they used the flowchart and text files while working on their assignments because those files were easier to use to find specific information quickly instead of scanning through the video or audio.

Qualitative Analysis

After both assignments, students were given a credit to complete a post-assignment survey. This survey, seen in Appendix F, included quantitative questions, which have already been covered, and some qualitative questions. There was a qualitative question for each type of material asking students for comments. There was also a question asking the students if they completed the example assignment online, why they did so, and a few other questions. To analyze the qualitative data, themes were coded and collected. A colleague also coded themes, and the results were compared and consolidated.

Flowchart Qualitative Data

Themes found while coding the comments about the flowcharts were: general comments about the flowcharts helping (in addition to helping with logic), organizing thoughts, understanding the program, and visualizing the program, along with knowing the order of tasks in the program (see Table 41).

Table 41: Themes. Flow Chart Qualitative Comments

Flow Chart Comments		
Themes	Assignment 8	Assignment 9
Helped logic	2	2
Helped	10	26
Clearly show program	2	1
Helped organize my thoughts	4	8
Knew the order of tasks	8	8
Helps to understand program	3	5
Helps to visualize	5	3

Audio Qualitative Data

Themes found while coding the comments about the audio files were not as supportive. Some students could not concentrate on the audio while other students said they could concentrate more on the audio than other materials. Other students complained that it was not visual enough to help and, more generally, that they listened to the audio and it did not help. In Assignment 9, several complained that the audio took too much time. Some students, eight on Assignment 8 and 12 on Assignment 9, said, generally, that it did help (see Table 40).

Table 42: Themes. Audio Qualitative Comments

Audio Comments		
Themes	Assignment 8	Assignment 9
Can pay attention to audio more	1	0
Not visual	8	10
Could not concentrate	5	2
Helped	8	12
Listened, didn't help	3	3
Takes more time	2	10

Text Qualitative Data

Observation of the comments about the text files, there were good and bad comments (see Table 43). Some of the good comments stated that information was easy to find by skimming. Students also commented that the text file helped them to understand the assignment and helped to explain the assignment along with just general comments that the text file helped. Some students complained that the text file was too long and that it was not visual enough. Other students commented that the text file contained a lot of information. This last comment could be negative or positive – it is difficult to tell without context.

Table 43: Themes. Text Qualitative Comments

Text Comments		
Themes	Assignment 8	Assignment 9
A lot of info	4	1
Easy to skim	1	1
Easy to find specific info	6	9
Helped	21	25
Helped to understand	3	4
Helped explained	2	0
Was too long	5	4
Was not visual	2	4

Example Assignment - Qualitative Data

Several students stated that the example assignment was generally helpful. Others were more specific about how the example was helpful; they could go back to review part of the example that was difficult to understand, it could help them more easily visualize the program, and it helped them understand the process because each step was walked through with the instructor (see Table 44).

Table 44: Themes. Example Assignment Qualitative Comments

Example Comments		
Themes	Assignment 8	Assignment 9
Helped	7	16
Could go back	5	4
Easy to visualize	1	0
Help understand the process	0	1

Choice of Out of Class Venue for Example Assignment

Students had their choice of working on the example assignment in or out of class. Observing the quantitative data about self-regulated students and preferred learning mode, there was not a clear relationship between either of those options. If there did happen to be a relationship, a qualitative question was included to try to find out this information.

Themes that were found in these comments included students that found programming easy and could complete the example easily on their own in a shorter amount of time (did not waste time while the instructor was helping students in class). It also included students that had a more difficult time with programming, but they could still get the material from the instructor and replay parts that were harder for them to understand (see Table 45). This course met at 8:00 a.m., so some students wanted to sleep in and the last explanation students gave was that it helped with their schedule. Some of the more defined ways that it helped their schedule was for athletes that had games out of town, students who had family emergencies, and that they could work at their own pace so they could work on their other studies.

Table 45: Themes. Choice of On-Line Venue for Example Assignment

Chose of On-Line Venue for Example Assignment		
Themes	Assignment 8	Assignment 9
Easy to go back	2	1
Helps my schedule	5	9
Wanted to sleep late	1	9
Programming is easy for me	2	3

Watched Example More than Once

Some students watched the video more than once. Anticipating that this may happen, a qualitative question that was included asked students why they watched more than once if they did. Most responses were that they went back and would watch it again to go over a harder part of the program because they did not understand it the first time (see Table 46).

Table 46: Themes. Reason for Watching Example More than Once

Watched Example More than Once		
Themes	Assignment 8	Assignment 9
Re-watched Difficult Part	3	2

Another student commented that the example was even more helpful by accessing the worked example explanation online and reviewing the example while working on the assignment. The student had multiple monitors and would have the example playing on one monitor while working on the other monitor. It was easy for the student to follow along while coding; they could stop the video when they needed to think about what was just covered, and could also take a break if needed.

Chapter Five – Discussion

Summary of Chapter Five

This research compared preferred learning modes of students in an introductory programming course. The preferred learning mode for each student was discovered by completing the VARK® survey (Neil D Fleming, 2011). This study looked at several different factors: the order students chose various teaching modes, the rankings the students gave to the preferred and most helpful of each teaching mode, and the assignment scores of different learning modes. By recording the example assignment while working through the example in class, students had the option to complete the example assignment in class or online while watching the recording. This research also calculated students' self-regulation score using the MSLQ (Pintrich, 1991), analyzing if this score is significant between students that worked on the example assignment on their own versus completing the example assignment in class. In other words, is a student's self-regulation score related to which venue they choose to complete the example assignment?

Students' Ranking of Teaching Modes (Preferred and Helpful)

The best scenario would be to identify students' best learning mode and matching them to instructional strategies that align with that teaching mode (Dunn & Dunn, 2016). Several students' rating of most helpful and most preferred materials for both assignments did not match their preferred learning mode.

There was a large difference between Assignment 8 and Assignment 9 on preferred and most helpful teaching mode for visual teaching mode (see Table 11). In Assignment 8 the visual, flowchart, was ranked most preferred and most helpful by two students. In Assignment 9, the visual, flowchart, was ranked most preferred and most helpful by eight students. In the post assignment survey, when asked for general comments about each teaching mode, several students commented that the flowchart was difficult to understand at first but became more helpful after they figured out how to read a flowchart. Students may have needed the experience with Assignment 8 to understand what teaching materials would help them more on Assignment 9. Assignment 9 programming assignment included more user interactions, between the person and the program, to achieve the solution (see Appendix E: Assignments and Rubrics). Because of the extra interactions, Assignment 9 program may have been harder than previous assignments for students to understand. The flow chart would help understanding of these extra interactions.

Previous research is lacking on discovering preferred learning mode with Neil D Fleming (2011) VARK® questionnaire and giving students options of teaching mode materials. Research by Buch and Sena (2001) reviewed different teaching modes materials with students that had taken the Kolb Learning Styles Inventory (D. Kolb, 1985), to find their preferred learning mode. The research included delivering four different sections of the course in Kolb's different teaching modes.

Students were then asked how enjoyable, their perceived learning, and how well students completed a knowledge post-test. Comparing these aspects with the students' learning mode, Buch and Sena (2001) found a statistically significant difference in lesson enjoyment and perceived learning but not on post-test knowledge scores.

This contradicts the findings because there was not a statistically significant difference, denoting a relationship between enjoyment and students' learning mode. Although this research used Neil D Fleming (2011) VARK® questionnaire to find preferred learning mode, and Buch and Sena (2001) used Kolb's Learning Style Inventory, both studies compared preferred and most helpful with preferred learning mode. Buch and Sena (2001) suggest that the discrepancy between post-test knowledge scores and lesson enjoyment/perceived learning may be due to customizing teaching modes and may give students more confidence. They suggest that learning this type of teaching may be most helpful in introductory courses. This is an introductory course, so this may be the reason there was a difference between ratings of teaching modes versus preferred learning mode from Neil D Fleming (2011) questionnaire. Self-reported preferred way of learning is often a bad predictor of the way people learn more effectively. What a person prefers, is not always what is best for them (Kirschner, 2017).

Assignment Scores by Preferred Learning Modes

In this research, there was not a statistical significant difference in learning modes within the class and their assignment scores. Previous research has had differing results.

Kim et al. (2015) gave their participants the VARK® questionnaire, (Neil D Fleming, 2011), and compared their scores from the American Board of Surgery In-Training Examination (ABSITE) by their VARK® questionnaire results. Their research suggested that students with a reading/writing preferred learning mode performed better on the ABSITE test, which consists solely of reading/writing mode.

Leung et al. (2014) compared students that passed micro- and macro-economics courses. Students in both courses first completed the VARK® questionnaire, (Neil D Fleming, 2011), then the students' final percentage grade for the course was compared by their preferred learning modes. There was not a statistically significant difference for the microeconomics course and that the kinesthetic learning mode had a significant positive relationship with the total percentage grade. The researchers suggested that students enjoyed a variety of teaching modes, and knowing students' preferred learning mode allows the instructor to personalize the instructional material used in the course.

Lujan and DiCarlo (2006) conducted research with their medical students. These students were administered the VARK® questionnaire, (Neil D Fleming, 2011). Although most students could learn as long as the instructor provided a blend of teaching modes, some of the students learned better in a specific mode and could not learn the material unless their preferred learning mode was the way the teaching material was presented.

Sankey et al. (2011) conducted a research project offering different teaching modes for different groups in the course. In this research, students were given a pre-test and post-test (the same test for each) to calculate learning achieved during the research project. Each group completed different teaching content for each concept. It was noted by the researchers that the learning concepts used were not difficult and the increased knowledge could have resulted from making logical assumptions or an intelligent guess. Like this research there was not a statistically significant difference in the score between groups and concepts (preferred learning modes and content's preferred teaching mode). This suggests that no relationship exists between the two groups.

Assignment Scores by Venue Used to Complete Example Assignment

Scores for students that chose to complete the example assignment online were higher in assignment eight and those that attended class to complete the example assignment for assignment nine were slightly higher. Neither of these were statistically significant. With assignment scores for students that completed the example online being higher, it suggests that overall this option does not hurt the students' scores and gives them the ability to review the recorded example assignment solution while they are working on their actual assignment.

The bigger difference in the mean of the students in assignment eight, higher for students that completed the example assignment online, may be explained by Schraw, Crippen, and Hartley (2006) suggestion about vicarious learning. Vicarious learning is when individuals learn by observing experts doing a task. When vicarious learning is taking place, students feel that while they are completing the task, they are not expected to complete the task on their own, so they can concentrate on paying attention to the expert, the instructor.

Some of the qualitative responses from the students in the post-assignment survey included that the student chose to complete the example assignment online because it took less time to complete. Instead of taking 50 minutes to complete the example, they could complete the example assignment in a much shorter time because the student did not have to wait while the instructor went around the classroom helping students that had errors.

The number of students involved and only collecting data from two assignments could affect the reliability of the data. Also, letting the students choose and not randomly assigning students to the groups could affect the results.

Venue for Completing Example Assignment by Preferred Learning Mode

The learning mode of working the example assignment is kinesthetic learning mode. A bigger percentage of students with kinesthetic preferred learning mode attended class in comparison to the other learning modes but not enough to suggest an association. The number of students that attended class with a preferred learning mode of kinesthetic may be high due to the high number of students with the kinesthetic learning mode. The low number of students in the course could cause the data to not be reliable, but without finding research with this same method, there is no other research to compare the findings.

Self-Regulated Score by Venue

In this research, self-regulation scores were calculated to see if the students that completed the example online were more of self-regulated students; i.e., students that do for themselves and take the initiative to learn on their own, but chose to work on the example online and made this choice due to being more proactive. Exploring self-regulation score shows no significant difference in students that attended class or completed the example online, when given the option. In other words, the results do not suggest that the students' self-regulation score is not related with students choosing to come to class or completing the example online.

Problems with Learning Styles and Assessment

This research did not find a significant difference in any of the calculations dealing with learning styles. After reviewing previous writers, Dembo and Howard (2007) looked at three questions to rate the quality of learning style assessment:

1. Are learning style instruments valid and reliable?
2. Do students benefit when the type of instruction matches their preferred learning style?
3. Is there evidence that understanding a student's learning style improves concentration, memory, self-confidence, and reduces anxiety which leads to better grades?

Validity is whether the instrument tests what it is supposed to test. These assessments are not decided by technical or statistical assessment. This brings arguments from some researchers that there are other dimensions that should be used when deciding if the instrument is valid. Learning style instruments use forced-choice questionnaires. Sometimes, one of the choices offered may be an option that most people would choose. For example, most students would prefer a demonstration instead of or along with a lecture, even if students do not have a preferred mode of visual. Therefore, this type of question/answer would have little value in showing a distinctive characteristic of the learner (Dembo & Howard, 2007). The scoring of the evaluation does not take into consideration the context of questions such as these.

Reliability concerns are whether the instrument will consistently produce the same or similar results when used repeatedly over time. If a theory behind a model suggests that the style may change over time or in different situations, it would be hard to say the model would be reliable (Dembo & Howard, 2007). Stahl (1999) states that the instruments may also not be reliable due to students not understanding the question because of reading issues.

Application, establishing the usefulness of learning styles, is probably the most important hurdle (Dembo & Howard, 2007). Stahl (1999) failed to find matching teaching pedagogy to learning styles to improve learning. With the various studies on learning styles, there is not a consistent and measurable improvement on learning. Without consistent results, application is not evidenced (Dembo & Howard, 2007).

Coffield, Moseley, Hall, and Ecclestone (2004) reviewed several learning style models. They state that some of the leading theorists make extravagant claims for their model, which reflects badly on the whole field. Others have an absence of sound evidence to base their arguments on - either anecdotal or 'implicit' suggestions, although, in their research, they suggest that a knowledge of learning styles can be used to increase self-awareness of students and tutors about the strengths and weaknesses of their learners.

Matching students' learning styles with teaching style, which is often recommended, is also just as often said to be unrealistic (Coffield et al., 2004). In this research, student's ranking of teaching materials used while working on their homework, may not have matched their preferred learning mode, but gave the students options. If more assignments were used in the study, research may see a student using different teaching materials for different concepts or trying different modes to find which one helps them complete the assignment easily and/or more accurately. The extra course material, teaching mode material to match each learning mode, did take a little extra time to put together. For a subject like programming that does not have changes often, this material will not need to be recreated each semester.

Implications

Programming Courses

One problem beginning programming students have is transferring the knowledge of what the program needs to achieve to the correct steps in the program (Lahtinen et al., 2005; Winslow, 1996). Sharp and Schultz (2013) researched using video in a programming course. They created videos of completing exercise programs. They did not discuss scaffolding while completing these exercise programs. During their research, they found that students spent more time watching the videos than they did reading the book (self-reported) and that students preferred watching the videos to reading the book, but neither were a significant difference. Not only did the students prefer watching the video to reading the book, the students also said the videos were more helpful than the book in

some topics. In their study, the instructor created the videos outside of class so they took extra time. Sharp and Schultz (2013) decided to continue including the video examples, even though they took extra time, because of the impact the videos had on their students.

Although video was a part of the current study, the content is more important than it being a video. In the current study, the instructor walked through completing an example assignment. Therefore, instead of just completing a programming exercise, the instructor talked through programming the example by asking questions and explaining in the class what should be done next and concepts that should be used. These videos did not take extra time because they were recorded while working through the example in class and posted afterwards. Sharp and Schultz (2013) suggested researching using videos with learning styles in a computer programming course because they did not include this in their study, which we did here.

A flowchart is most often used to visually show steps needed to complete a task. Flowcharts were posted in each assignment for visual learners. In the students' ranking of most helpful material in assignment nine, the flowchart option was rated the second most helpful, only behind the example assignment. Only one of the eight students that chose flowcharts as the most helpful in assignment nine, had a preferred learning mode of visual. This suggests that even if an instructor is not interested in learning modes, including a flowchart in programming courses may help the students complete their assignments. Observation of the qualitative data, the flowcharts could help with the transfer discussed earlier (converting the idea of the solution to code). Another reason to include flowcharts is that even though student programmers know how to solve the problem, they have trouble converting it to a program (Winslow, 1996). In addition, students support this with the comments on the post-assignment survey. From the themes earlier, 1) helped with the logic, 2) helped organize my thoughts, 3) helped with the order of the tasks, 4) helped to understand the program, and 5) helped to visualize the program.

The example assignment was overwhelmingly rated as the most helpful, in both assignments, which substantiates the fact that students learn best with worked examples (Alfieri et al., 2011; Caspersen & Bennedsen, 2007; Jonassen, 2003, 2004; Lahtinen et al., 2005; Sweller et al., 1998), and experiential learning (D. A. Kolb, 1984), and video (Sharp & Schultz, 2013; Whatley & Ahmad, 2007). Working an example is helpful, but a video of working that example can be even more helpful for students, because they can review it while working on their assignment.

Instructional Design

The results in this study support Kirschner (2017) sentiment that learning modes are not a main factor when it comes to pedagogy for a course. This study did not find a relationship in any comparisons when it came to instructional design and learning modes.

However, as seen in the numbers of most helpful type of teaching materials, each type was chosen more than once. Even though the preferred learning mode may not be the same teaching mode as rated most helpful, students still felt each teaching material was helpful. Although learners' preferred method of learning is not always the best way for them to learn (Kirschner, 2017), but whenever possible, maybe different modes of teaching materials should be posted for students in a beginning programming course. When a beginning programming course is designed for a variety of students' interests and backgrounds, students can be more motivated and have less anxiety towards the course (Bosch et al., 2011; Brosnan et al., 2010; Forte & Guzdial, 2005; Gilroy & Desai, 1986; Igbaria & Chakrabarti, 2007). This could also help in other types of courses where information does not change frequently since it takes extra time to create the extra teaching materials.

In this study, even though students did have material that matched their preferred learning mode, most students did not choose the material that matched their learning mode first. Most also did not rate as most helpful while working on their assignment the same teaching mode as their preferred learning mode. If this is the case, the reason to offer alternate types of learning materials should not be to match the students' preferred learning style, but just to offer students additional learning materials.

Caspersen and Bennedsen (2007) found that worked examples should be the foundation of instructional design to any programming course and should not be static because the programmer must design a mental model of the solution. Working the example and having the students work along meets this criterion. Working through this example and having a completed example program before working on the actual assignment, allows the student to put this example, and knowledge from working the example along with the professor, in their personal 'library' for future use (Jonassen, 2004).

Learning Styles

Validity, reliability, and predictive powers are all issues with learning style assessments (Dembo & Howard, 2007; Kirschner, 2017; Stahl, 1999). Reliability looks at whether a learner will answer the questions from the learning mode assessment the same at different times. Test/retest reliability is low (Kirschner, 2017). Most of the learning style assessments are self-reported. Learners are either not able or are not willing to honestly report what they do or what they think they do (Kirschner, 2017).

Kirschner (2017) discussed that the problem with learning styles is the lack of scientific basis. Offering students teaching modes to match their learning mode does not guarantee that the student will learn optimally. The learner is pigeon-holed into a style or group without using scores on several dimensions. Any differences would be gradual and not nominal, chunked into groups.

In this study, none of the comparisons with learning modes showed a statistically significant correlation, or relationship between the variables. From Kirschner (2017), research, data may not have shown any relationship because they used the VARK® questionnaire (Neil D Fleming, 2011) to evaluate the learning mode for each student. This questionnaire asks learners how they feel they understand when information is presented to them. Therefore, this is a self-reporting survey where the learners may not be able to accurately choose the correct answer for that individual for each situation.

Limitations

This course had a small population; for a 95% confidence level and an unlimited number in the population, research needs 385 or more participants. That could have altered the results. Not only did the small population affect some of the statistical calculations, with a larger population, a control group could be included.

Another aspect has to do with the number of assignments used in the study. Gathering data for more than two assignments could also alter the results by making them more reliable.

Tegrity was used to record the walk-through of the example assignment with the students because that was the recording software used by our University at the time. Blackboard was used as the content management software for the same reason. The way Blackboard and Tegrity work together, the students could have issues when accessing Tegrity if the link to Tegrity is included on the page where assignment material is posted in Blackboard. Therefore, students had to click on the Tegrity tab on the left side of the screen to access the example assignment walk-through online (see Appendix I). This could affect the results of which order the students accessed the materials because all the links to materials were in one place except for the example walkthrough.

In addition, the assignment description was posted in the same area as the extra materials. This could also have affected the order in which the materials were accessed. In an ideal study, all the extra materials should be posted in one area with no other material posted there.

Content of Additional Teaching Mode Materials

All the extra material was in a different place, so the setup of the course in Blackboard was not authentic. Normally, an instructor would put all the material in one folder. The folder with the extra material did not contain the reading links and the lecture slides. Also, with all this material in a different place than the reading and lecture slides, it might have affected students' use of these different teaching mode materials.

Even though all the additional material posted were worth 5 points each, I neglected to put the points in the text of the material in Blackboard, else it could have altered the viewing of the material.

The additional teaching mode material was not reviewed by anyone other than the instructor. To verify that the material is easy to use and understand, more people than just the instructor, including instructional designers, subject matter experts, and individuals that are not familiar with the material, should review it.

The use of the materials while working on the assignment and/or the content for each of the teaching mode materials did not contain the same material. The text and audio were very similar and contained the same content. By description of the flowchart and working through the example assignment, their content was much different than the text and audio files. Some teaching modes are not good for a specific task, whereas some modes are better for specific tasks. For instance, instructions of how to change a tire may be best understood by most people, without regard to their preferred learning mode, by a video or someone actual leading you through the process, kinesthetic, instead of just hearing or reading the instructions, auditory or read/write.

Venue of Completing Example Assignment

Allowing students the choice to complete the example online or in class allowed collection of data to test whether students with higher self-regulated scores would choose a specific venue. However, not having the students randomly assigned to those groups could affect the results of assignment scores compared to venue of completing the example assignment.

Prior Experience

Students' prior programming/computer experience could affect the results, but were not taken into consideration with the calculations in this research. Previous experience could affect the order the materials were viewed. With more experience, some materials could be chosen to be viewed first.

In addition, this experience could affect grades on the assignments. Attending class or completing the example assignment on-line could also be affected by previous programming experience. Some students that left comments about working on the example assignment on-line were those who understood the material on their own. These students had previous programming experience even if it was in a different programming language (see Table 45).

Delimitations

It is important to remember that this study was conducted with students at a higher education level. Repeating the study with students of a different age group may alter the results. These students were also enrolled in a required programming course. If a programming course is not required, the results may vary with students opting to take the course, which can affect their motivation and interest, instead of being required to take the course.

Future Research

Completing a similar study with more participants and assignments. Including assignments for an entire semester. With more participants, the students can be divided

into two random groups allowing one group to receive the extra materials and one group that does not have access to the additional material.

Another study may include having materials like these, randomly dividing the students into two groups and alternate between assignments with one group having access to the materials and the other group not having access to the materials.

In this research, data was gathered about which material was viewed first, but qualitative data was not collected about why they chose each type of learning material. Asking the students why they chose the order of reviewing each piece of learning material may give more insight as to why students did not choose the material that matched their preferred learning mode first. Besides which material was chosen first, students were asked which material was most helpful and preferred. Most of those selections did not match the students' learning mode either, so asking why they rated the most helpful material and why each of those were chosen could be informative also.

Offering a variety of materials may affect how students feel about their instructor. If the instructor is willing to provide multiple forms of learning documents, the students may feel that the instructor cares more and may like the instructor better. These feelings of student to instructor may also flow over into how they feel about the materials and course as a whole. Doing another research similar to this one but to also collect data about how students feel about their instructor may reveal more information about how students rate the additional materials.

Not only could offering these additional teaching mode materials make the students feelings toward the instructor change, it may also improve their motivation.

Conclusions

Designing courses to align with students preferred learning modes is a common recommendation, yet there are disagreement regarding whether learning modes and their impact on learning can be evaluated scientifically. This research did not show that providing teaching materials to match each learning mode in the VARK® model affected students' scores on their assignments or material preferred by the students. Although preferred learning modes did not affect these aspects of the course, students enjoyed and utilized the extra teaching materials.

When giving students a variety of ways to complete an assignment or task, the instructor should be aware that students might choose the option that requires less effort for no other that reason. Although some students gave an answer that seemed to have more responsible than “because I didn’t want to go to class” or “I didn’t want to get up early” the underlying reason may have been because it took less effort to work on the example on-line.

After completing this research, I do not believe that learning modes are important, at least not for an introductory programming course. Several of the teaching material types are just not a good fit when learning to program. For instance, just an audio file of information with no visual is hard for students to follow and understand Learning mode surveys are rated by self-reporting which is not a valid, scientific format for collecting data. In addition, what people say is not actually how they react. One more self-reported problem is that what people choose may not be the best response for them. People sometimes say something different. Because students enjoyed and utilized the additional

teaching materials, giving students more types of teaching materials may help students be successful in an introductory programming more than worrying about learning modes.

References

- Alfieri, L., Brooks, P. J., Aldrich, N. J., & Tenenbaum, H. R. (2011). Does Discovery-Based Instruction Enhance Learning? *Journal of Educational Psychology*, 103(1), 1-18.
- Bjork, R. A., Dunlosky, J., & Kornell, N. (2013). Self-Regulated Learning: Beliefs, Techniques, and Illusions. *Annual Review of Psychology*, 64, 417-444. doi:10.1146/annurev-psych-113011-143823
- Boekaerts, M. (1999). Self-regulated Learning: Where we are Today. *International Journal of Educational Research*, 31, 445-457.
- Bosch, N., D'Mello, S., & Mills, C. (2011). What Emotions Do Novices During their First Computer Programming Learning Session? *Artificial Intelligence in Education*.
- Brosnan, M., Gallop, V., Iftikhar, N., & Keogh, E. (2010). Digit ratio (2D:4D), academic performance in computer science and computer-related anxiety. *Personality and Individual Differences*, 51, 371-375.
- Buch, K., & Sena, C. (2001). Accomating Diverse Learning Styles in the Design and Delivery of On-line Learning Experiences. *The International Journal of Engineering Education*, 17(1), 93-98.
- Caspersen, M. E., & Bennedsen, J. (2007). Instructional Design of a Programming Course -- A Learning Theoretic Approach. *ACM*, 111-122.
- Chang, S. E. (2005). Computer Anxiety and Perception of Task Complexity in Learning Programming-Related Skills. *Computers in Human Behavior*, 21, 713-728. doi:10.1016/j.chb.2004.02.021
- Coffield, F., Moseley, D., Hall, E., & Ecclestone, K. (2004). *Learning styles and pedagogy in post-16 learning: a systematic and critical review*. Retrieved from London:
- Csorny, L. (2013). Careers in the growing field of information technology services. Retrieved from <https://www.bls.gov/opub/btn/volume-2/careers-in-growing-field-of-information-technology-services.htm>
- Dembo, M. H., & Howard, K. (2007). Advice about the Use of Learning Styles: A Major Myth in Education. *Journal of College Reading & Learning*, 37(2), 101-109.
- Dou, Y., & Shi, Q. (2015). *Research On Teaching Mode Practice In The Course Of Programming Based On CDIO*. Paper presented at the 11th International CDIO Conference, P.R. China.
- Dunn, R., & Dunn, K. (2016). Learning Styles, Teaching Styles. *NASSP Bulletin*, 59(393), 37-49.
- Felder, R. M. (Producer). (2010, 3/8/2015). ARE LEARNING STYLES INVALID? (HINT: NO!).
- Felder, R. M., & Brent, R. (2005). Understanding Student Differences. *Journal of Engineering Education*, 94(1), 57-72.

- Fleming, N. D. (1995). I'm different; not dumb. Modes of presentation (V.A.R.K.) in the tertiary classroom. *Research and Development in Higher Education, Proceedings of the 1995 Annual Conference of the Higher Education and Research Development Society of Australasia (HERDSA)*, 18, 308 - 313.
- Fleming, N. D. (2011). VARK: A Guide to Learning Styles. 7.1.
- Fleming, N. D. (2016). VARK: A Guide to Learning Styles. *Research and Statistics*. Retrieved from <http://vark-learn.com/introduction-to-vark/research-statistics/>
- Fleming, N. D., & Baume, D. (2006). Learning Styles Again: VARKing up the right tree! *Educational Developments*(7.4), 4-7.
- Fleming, N. D., & Mills, C. (1992). Not Another Inventory, Rather a Catalyst for Reflection. *To Improve the Academy*, 11, 137-155.
- Forte, A., & Guzdial, M. (2005). Motivation and Nonmajors in Computer Science: Identifying Discrete Audiences for Introductory Courses. *IEEE Transactions on Education*, 48(2), 248-253.
- Gao, R. (2011). *Reforming to Improve the Teaching Quality of Computer Programming Language*. Paper presented at the The 6th International Conference on Computer Science & Education (ICCSE 2011).
- Gilroy, F. D., & Desai, H. B. (1986). Computer anxiety: sex, race, and age. *International Journal of Man-Machine Studies*, 25(6), 711-719.
- Gomez-Albarran, M. (2005). The Teaching and Learning of Programming: A survey of Supporting Software Tools. *The Computer Journal*, 28(2), 130-144. doi:10.1093/comjnl/bxh080
- Hall, T. S. (2007). *Improving Self-efficacy in Problem Solving: Learning from Errors and Feedback* ProQuest.
- Hawk, T. F., & Shah, A. J. (2007). Using Learning Style Instruments to Enhance Student Learning. *Decision Sciences Journal of Innovative Education*, 5(1), 1-19.
- Hawthorne, K., Prout, H., Kinnersley, P., & Houston, H. (2009). Evaluation of different delivery modes of an interactive e-learning programme for teaching cultural diversity. *Patient Education and Counseling*, 74, 5-11. doi:10.1016/j.pec.2008.07.056
- Igbaria, M., & Chakrabarti, A. (2007). Computer Anxiety and Attitudes Towards Microcomputer Use. *Behaviour & Information Technology*, 9(3), 229-241.
- Jonassen, D. H. (2003). Designing Research-Based Instruction for Story Problems. *Educational Psychology Review*, 15(3), 267-296.
- Jonassen, D. H. (2004). *Learning to Solve Problems: An Instructional Design Guide*: Pfeiffer.
- Kim, R. H., Gilbert, T., & Ristig, K. (2015). The Effect of Surgical Resident Learning Style Preferences on American Board of Surgery In-Training Examination Scores. *Journal of Surgical Education*, 72(4), 726-731.
- Kirschner, P. A. (2017). Stop propagating the learning styles myth. *Computers & Education*, 106(2017), 166-171.
- Ko, A. J., Myers, B. A., & Aung, H. H. (2004). *Six Learning Barriers in End-User Programming Systems*. Paper presented at the Proceedings of the 2004 IEEE Symposium on Visual Languages and Human Centric Computing, Rome, Italy.

- Kolb, A. Y., & Kolb, D. A. (2005). Learning Styles and Learning Spaces: Enhancing Experiential Learning in Higher Education. *Academy of Management Learning & Education*, 4(2), 193–212.
- Kolb, A. Y., & Kolb, D. A. (2008). Experiential Learning Theory: A Dynamic, Holistic Approach to Management Learning, Education and Development. *Armstrong: Management Learning, Edu. and Develop*, 1(20), 42-68.
- Kolb, D. (1985). Learning Styles Inventory *The Power of the 2 x 2 Matrix: Using 2 x 2 Thinking to Solve Business Problems and Make Better Decisions* (pp. 267-271): Jossey-Bass.
- Kolb, D. A. (1984). *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Cliffs, NJ: Prentice Hall.
- Lahtinen, E., Ala-Mutka, K., & Harvinen, H.-M. (2005). A Study of the Difficulties of Novice Programmers. *ACM SIGCSE Bulletin*, 37(3), 14-18.
doi:10.1145/1067445.1067453
- Leung, A., McGregor, M., Sabiston, D., & Vriliotis, S. (2014). VARK Learning Styles and Student Performance in Principles of Micro- vs. Macro-Economics. *Journal of Economics and Economic Education Research*, 15.3, 113-120.
- Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hammer, J., Lindholm, M., . . . Thomas, L. (2004). A Multi-National Study of Reading and Tracing Skills in Novice Programmers. *ACM*, 36(4), 119-150. doi:10.1145/1041624.1041673
- Lujan, H. L., & DiCarlo, S. E. (2006). First-year medical students prefer multiple learning styles. *The American Physiological Society*, 30, 13–16.
- McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-D., . . . Wilusz, T. (2001). A Multi-National, Multi-Institutional Study of Assessment of Programming Skills of First-Year CS Students. *ACM SIGCSE Bulletin*, 33(4), 125-180. doi:10.1145/572139.572181
- McKeachie, W. J., Pintrich, P. R., & Lin, Y.-G. (1985). Teaching Learning Strategies. *Educational Psychologist*, 20(3), 153-160.
- Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., . . . Paterson, J. (2007). A Survey of Literature on the Teaching of Introductory Programming. *ACM SIGCSE Bulletin*, 39(4), 204-223.
- Pintrich, P. R. (1991). A Manual for the Use of the Motivativated Strategies for Learning Questionnaire.
- Pintrich, P. R., & De Groot, E. V. (1990). Motivational and Self-Regulated Learning Components of Classroom Academic Performance. *Journal of Educational Psychology*, 82(1), 33-40.
- Pintrich, P. R., McKeachie, W. J., & Lin, Y.-G. (1987). Teaching a Course in Learning to Learn. *Teaching of Psychology*, 14(2), 81-85.
- Purdie, N., Hattie, J., & Graham, D. (1996). Student Conceptions of Learning and Their Use of Self-Regulated Learning Strategies: A Cross-Cultural Comparison. *Journal of Educational Psychology*, 88(1), 87-100.
- Ramalingam, V., LaBelle, D., & Wiedenbeck, S. (2004). Self-efficacy and Mental Models in Learning to Program. *ACM SIGCSE Bulletin*, 36(3), 171-175.

- Reed, S. K., & Bolstad, C. A. (1991). Use of Examples and Procedures in Problem Solving. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 17(4), 753-766.
- Reid, J. M. (1987). The Learning Style Preferences of ESL Students. *Teachers of English to Speakers of Other Languages, Inc.*, 21(1), 87-111.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137-172.
- Rosenberg, M., & Burkert, A. (2015). Learning Styles and their Effect on Learning and Teaching *Forschende Fachdidaktik* (pp. 103-130): Francke Verlag.
- Rountree, N., Rountree, J., Robins, A., & Hannah, R. (2004). Interacting Factors that Predict Success and Failure in a CS1 Course. *Inroads - The SIGCSE Bulletin*, 36(4), 101-104.
- Rowe, F. A., & Rafferty, J. A. (2013). Instructional Design Interventions for Supporting Self-Regulated Learning: Enhancing Academic Outcomes in Postsecondary E-Learning Environments. *Journal of Online Learning and Teaching*, 9(4), 590-601.
- Sankey, M. D., Birch, D., & Gardiner, M. W. (2011). The impact of multiple representations of content using multimedia on learning outcomes across learning styles and modal preferences. *International Journal of Education and Development using Information and Communication Technology*, 7(3), 18-35.
- Schraw, G., Crippen, K. J., & Hartley, K. (2006). Promoting Self-Regulation in Science Education: Metacognition as Part of a Broader Perspective on Learning. *Research in Science Education*, 36, 111-139.
- Sharp, J. H., & Schultz, L. A. (2013). An Exploratory Study of the use of Video as an Instructional Tool in an Introductory C# Programming Course. *Information Systems Education Journal (ISEDJ)*, 11(6), 33-39.
- Stahl, S. A. (1999). Different Strokes for Different Folks? A Critique of Learning Styles. *American Educator*, 23(3), 27-31.
- Sweller, J., van Merriënboer, J. J. G., & Paas, F. F. W. C. (1998). Cognitive Architecture and Instructional Design. *Educational Psychology Review*, 10(3), 251-296.
- Tanner, K., & Allen, D. (2004). Approaches to Biology Teaching and Learning: Learning Styles and the Problem of Instructional Selection—Engaging All Students in Science Courses. *Cell Biology Education*, 3(Winter), 197-201.
- Urval, R. P., Kamath, A., Ullal, S., Shenoy, A. K., Shenoy, N., & Udupa, L. A. (2014). Assessment of learning styles of undergraduate medical students using the VARK questionnaire and the influence of sex and academic performance. *Advances in Physiology Education*, 38, 216–220.
- Whatley, J., & Ahmad, A. (2007). Using Video to Record Summary Lectures to Aid Students' Revision. *Interdisciplinary Journal of Knowledge and Learning Objects*, 2, 185-196.
- Winslow, L. E. (1996). Programming Pedagogy -- A Psychological Overview *SIGCSE Bulletin*, 28(3), 17-25.
- Zimmerman, B. J. (1986). Development of a Structured Interview for Assessing Student Use of Self-Regulated Learning Strategies *American Educational Research Association*, 23(4), 614-628.

- Zimmerman, B. J. (1989). A Social Cognitive View of Self-Regulated Academic Learning. *Journal of Educational Psychology*, 81(3), 329-339.
- Zimmerman, B. J. (2000). Handbook of Self-Regulation *Handbook of Self-Regulation*: Academic Press.
- Zimmerman, B. J. (2002). Becoming a Self-Regulated Learner: An Overview *Theory Into Practice*, 41(2), 64-70.
- Zimmerman, B. J., & Schunk, D. H. (2001). Reflections on theories of self-regulated learning and academic achievement *elf-regulated learning and academic achievement: Theoretical perspectives* (Vol. 2, pp. 289-307).

Appendix A: Course Syllabus

IST 1551 – Information Systems: User Perspective
Course Syllabus – Spring 2016

Department Mission

“Capitalizing on the strong technological emphasis of Missouri S&T, the Department of Business and Information Technology educates individuals for careers in modern business organizations. The Department emphasizes management through technology with particular focus on information systems and their application in a fast-changing, global, and competitive environment. Through innovative instruction and research, the Department serves the economic interests of industry and the evolving needs of society.”

Instructor Information

Instructor: Professor Carla Bates *Class:* MWF 251 Toomey
Office: 105 Fulton Hall *Classroom:* 251 Toomey
Phone: 573-341-7719 *Office Hours:* M: Noon. – 2:00 p.m. (251 Toomey)
E-Mail: carla@mst.edu Tues: 10:00 a.m. -11:30 a.m. (on-line)
F: 10 a.m. – noon (105 Fulton)
And by appointment

Course Information

Catalog Description:

Introduction to object-oriented programming in the context of developing and implementing various components of an information system with particular attention given to system interface such as window and web forms. Class will include numerous projects covering foundational programming.

Extended Description:

We will complete programming assignments using Visual C# and Visual Studio 2013. Information systems are the focus of the Information Science and Technology department. An information system is a large set of programs and interfaces that provide timely data for business operations and decision making requirements. There are many important aspects of information systems common to most commercial implementation. In particular, IST 1551 focuses on the issues of implementation surrounding the system user interfaces.

Course Prerequisites:

None.

Required Materials:

None

Instructional Methods:

The instruction is a combination of lecture and laboratory. The lectures will be used to teach important concepts, definitions, and techniques. The example assignments and labs will be highly interactive and directed experiences allowing a hands-on approach to learning the programming exercises.

Course Learning Objectives:

Course Objectives	Program Learning Objectives				
	Oral Communication	Written Communication	Critical Thinking	Information Technology	Teamwork
Understand the various components of an information system and their role in a corporate organization	X	X	X	X	
Learn how to use the Visual C#.NET environment to build information system components		X	X		
Understand and utilize basics of object-oriented design and how it relates to .NET Framework			X	X	
Integrate databases into Visual C# programs using ADO.NET			X	X	
Master the core programming concepts with regards to the manipulation of data fields			X	X	
Learn to read simple programming requirement documents		X			

Course Assignments

Labs: Computer laboratory time will be made for each assignment. You are strongly encouraged to attend the labs if you have not already completed your assignment. It is also encouraged that you come to the lab class with the form completed so you can receive help with the more difficult part (programming). This is a great time to get one-on-one help with the assignment.

Homework:

(50 points each): Homework assignments will be made frequently. Even though there will be scheduled lab days, you will have to work on them outside of class. **DO NOT** wait until the day they are due to start them. A rubric will be posted for each assignment to help you receive the maximum points possible. They are expected to be completed separately. You are allowed to receive help from friends or classmates, but there **must** be differences (different filenames, object names on form, variables, etc.). If I have homework that is identical between students, **all students involved** may receive a 0 for that assignment and the Office of Undergraduate Studies will be notified (See **Academic Integrity Statement** below). Homework exercises are to be completed before the specified date and time and uploaded to Blackboard in the Assignments tab. Late homework will receive a 10% reduction for each day late with a maximum of up to 3 days late. **NO** assignments will be accepted more than 3 days late. A zero will be given for any assignment not posted within the deadline. **Please DO NOT ask after the 3 day grace deadline unless you have documented extenuating circumstances.**

Quizzes/Example Assignments (10 points each):

We will have regular quizzes and example assignments. Quizzes will be administered at the **start** of the class and will be multiple-choice. Students cannot start the quiz more than 5 minutes after the beginning of class. These will use the Blackboard assessment tool and your score will be posted immediately. They will be open notes and book but not open neighbor and must be taken in class that day. The quizzes will be over the previous lecture so having those slides available will be a great help.

Along with quizzes, we will also have some example assignments. The example assignments the first few weeks of class, will be completed in class and you must attend and submit to get the 10 points. Later in the semester, you will have other options that will be explained to you at that time.

Combining quizzes and example assignments, we will have more than 20, but only the best 20 will be used to calculate your grade (this will allow you to miss a few classes without your grade suffering).

Examinations: The midterm and final will consist of two parts, a debugging exercise and a programming exercise. When working in a business environment, you will be expected to debug your own code as well as other people's code so the debugging portion is important.

Course Policies and Grading**Evaluation Methods:**

Grades will be based on total points, as defined below. I will **NOT** round your final grade up at all if you miss **ANY** assignments. If you do complete every assignment/example assignment and most quizzes, I will be **generous** about rounding up. So, it will benefit you to attend class and complete **EVERY** assignment and example assignment.

I have added 5 extra credit points on every assignment. It is suggested that you take advantage of these extra credit points on each assignment. Please do not ask me at the end of the semester for extra credit opportunities. *Note: Remember in-department students, a C is a passing grade.*

Evaluation Methods: Assignments, daily quizzes/example assignments and examinations as seen below.

Assessment	Description	Points	Number	Total Points
Assignment 1	Introduction to C#	50	1	50
Assignment 2	Processing data	50	1	50
Assignment 3	Making decisions	50	1	50
Assignment 4	Loops	50	1	50
Assignment 5	Random numbers/Load event	50	1	50
Assignment 6	Methods	50	1	50
Assignment 7	Arrays	50	1	50
Assignment 8	Arrays, Lists	50	1	50
Assignment 9	More processing data	50	1	50
Assignment 10	Classes/Multiform	50	1	50
Examinations	Debugging Exams	50	2	100
	Programming Exams	100	2	200
Daily Quizzes / In-Class Assignments	Quizzes over the Reading Material and Lectures	10	20	200
				1,000

Grading Scales:

A: 100% - 90%

B: 89% - 80%

C: 79% - 70%

D: 69% - 60%

F: Below 59%

Attendance:

Attendance is strongly encouraged, the more you miss class, the more material will be foreign to you and more difficult to complete your assignments. You will likely want to ask questions. The class moves quickly and it is **easy to fall behind and not get caught up**. If emergency circumstances arise, please contact the instructor as soon as possible to avoid penalties, and to try to catch up to the rest of the class.

Course Outline:

The course will proceed as follows. This schedule is a best estimate. We may drift a bit from it.

It should be made clear that you are expected to work on the assignments outside of class. Pieces of the assignments will be worked during lectures and we will have some lab time for each assignment, but you will need time to bring them closure. Plan on a couple of hours outside of class for each hour spent in class.

Lecture Week	Topics
Jan 19	Preliminaries, Introduction to Visual Studio
Jan 25	Introduction to C#
Feb 1	Processing data
Feb 8	Making Decisions
Feb 15	Loops
Feb 22	Exception Handling
Feb 29	Mid-term
Mar 7	Random Numbers
Mar 14	Modularizing your code with Methods
Mar 21	Arrays
	SPRING BREAK
Apr 4	More on Arrays, Lists
Apr 11	More about Processing data
Apr 18	Classes and multiform projects
Apr 25	Continued from previous week
May 2	Final
Final	Programming/Debugging

Student Honor Code and Academic Integrity:

Please take a few minutes to stress the importance of academic integrity in class. Discuss why it should matter to the student, why it matters to you and your discipline, why it matters to Missouri S&T, and why it matters to future employers. Include a statement on your syllabus about the Honor Code developed and endorsed by the Missouri S&T Student Council: the Honor Code can be found at this link:

<http://stuco.mst.edu/about/honor.shtml>. Encourage students to read and reflect upon the Honor code and its emphasis on HONESTY and RESPECT. Page 30 of the Student Academic Regulations handbook describes the student standard of conduct relative to the University of Missouri System's Collected Rules and Regulations section 200.010, and offers descriptions of academic dishonesty including cheating, plagiarism or sabotage (<http://registrar.mst.edu/academicregs/index.html>). Additional guidance for faculty, including the University's Academic Dishonesty Procedures, is available on-line at <http://ugs.mst.edu>. Other informational resources for students regarding ethics and integrity can be found online at <http://ugs.mst.edu/academicintegrity/studentresources-ai>

S&Tconnect: <https://blackboard.mst.edu/> (S&Tconnect tab)

Coming fall 2014, Missouri S&T is implementing a new advising system as part of the four UM campuses Comprehensive Retention Initiative called S&Tconnect. S&Tconnect provides an enhanced system that allows students to request appointments with their instructors and advisors via the S&Tconnect calendar, which syncs with the faculty or staff member's Outlook Exchange calendar. S&Tconnect will also facilitate better communication overall to help build student academic success and increase student retention. S&Tconnect Early Alert will replace the Academic Alert system used by Missouri S&T. However, Academic Alert will continue to run in parallel with Early Alert until the end of the fall 2014 semester. Training will be provided beginning opening week of fall 2014 semester.

Classroom Egress Maps:

Faculty should explain where the classroom emergency exits are located. Please include a statement in your course syllabus asking the students to familiarize themselves with the classroom egress maps posted on-line at: <http://registrar.mst.edu/links/egress/>.

Disability Support Services: <http://dss.mst.edu>

Any student inquiring about academic accommodations because of a disability should be referred to Disability Support Services so that appropriate and reasonable accommodative services can be determined and recommended. Disability Support Services is located in 204 Norwood Hall. Their phone number is 341-4211 and their email is dss@mst.edu.

Instructors may consider including the following statement on their course syllabus as a means of informing students about the services offered:

"If you have a documented disability and anticipate needing accommodations in this course, you are strongly encouraged to meet with me early in the semester. You will need to request that the Disability Services staff send a letter to me verifying your disability and specifying the accommodation you will need before I can arrange your accommodation."

LEAD Learning Assistance <http://lead.mst.edu>

The Learning Enhancement Across Disciplines Program (LEAD) sponsors free learning assistance in a wide range of courses for students who wish to increase their understanding, improve their skills, and validate their mastery of concepts and content in order to achieve their full potential. LEAD assistance starts no later than the third week of classes. Check out the online schedule at <http://lead.mst.edu/assist>, using zoom buttons to enlarge the view. Look to see what courses you are taking have collaborative LEAD learning centers (bottom half of schedule) and/or Individualized LEAD tutoring (top half of the schedule). For more information, contact the LEAD office at 341-7276 or email lead@mst.edu.

The Burns & McDonnell Student Success Center

The Student Success Center is a centralized location designed for students to visit and feel comfortable about utilizing the campus resources available. The Student Success Center was developed as a campus wide initiative to foster a sense of responsibility and self-directedness to all S&T students by providing peer mentors, caring staff, and approachable faculty and administrators who are student centered and supportive of student success. Visit the B&MSSC at 198 Toomey Hall; 573-341-7596; <mailto:success@mst.edu>; Facebook: www.facebook.com/mstssc; web: www.studentsuccess.com

Title IX

Missouri University of Science and Technology is committed to the safety and well-being of all members of its community. US Federal Law Title IX states that no member of the university community shall, on the basis of sex, be excluded from participation in, or be denied benefits of, or be subjected to discrimination under any education program or activity. Furthermore, in accordance with Title IX guidelines from the US Office of Civil Rights, Missouri S&T requires that all faculty and staff members report, to the Missouri S&T Title IX Coordinator, any notice of sexual harassment, abuse, and/or violence (including personal relational abuse, relational/domestic violence, and stalking) disclosed through communication including but not limited to direct conversation, email, social media, classroom papers and homework exercises.

Missouri S&T's Title IX Coordinator is Vice Chancellor Shenethia Manuel. Contact her directly (manuel@mst.edu; (573) 341-4920; 113 Centennial Hall) to report Title IX violations. To learn more about Title IX resources and reporting options (confidential and non-confidential) available to Missouri S&T students, staff, and faculty, please visit <http://titleix.mst.edu>.

Appendix B: Consent Form

CONSENT FORM

Learning Styles

You are about to participate in a study of learning styles. The purpose of this study is to investigate learning styles/preferences, teaching styles, and self-regulated learners by learning styles.

For the study, you will be asked to fill out a survey at the beginning of the research to gather demographic and self-regulated learners. If you participate in the study, you will also be given the opportunity to review and complete the example assignment outside of class via recording. All participants will also be asked to complete a short survey after each assignment asking about viewing of the recorded example assignment.

Your participation is voluntary and your decision whether or not to participate will not affect any of your current and future relations with courses you take or the University of Missouri. Even if you decide to participate, you do not have to answer all the questions. You may discontinue participation at any time.

Questionnaire responses will be available only to the investigators of the study. Any information obtained in connection with this study will remain strictly confidential. All the identifying information will be removed after data have been collected.

If you are under 18 years of age, you cannot participate in this study. Should you have any questions, please feel free to contact Carla Bates by email at carla@mst.edu. For any other information regarding human participation in research, please feel free to contact the UM Campus IRB Office at 573-882-9585.

SIGNATURE

I confirm that the purpose of the research, the study procedures, the possible risks and discomforts as well as potential benefits that I may experience have been explained to me. Alternatives to my participation in the study also have been discussed. I have read this consent form and my questions have been answered. My signature below indicates my willingness to participate in this study.

Subject/Patient*

Date

Appendix C: Pre-Questionnaire

Q1

What is your age?

Q2 What previous programming experience do you have?

- ☐ No previous programming experience before this course
 - ☐ Self-taught programming (on your own, not as a course)
 - ☐ 1 programming course
 - ☐ 2 programming courses
 - ☐ More than 2 programming courses
- (if No is chosen, skip to Q4)

Q3

Was the programming experience you had in C#?

- ☐ Yes
- ☐ No

Q4

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

	1:Not at all true of me	2	3	4	5	6	7: Very true of me
During class time I often miss important points because I'm thinking of other things.	☐	☐	☐	☐	☐	☐	☐
When reading for this course, I make up questions to	☐	☐	☐	☐	☐	☐	☐

<p>help focus my reading.</p> <p>When I become confused about something I'm reading for this class, I go back and try to figure it out.</p> <p>If course readings are difficult to understand, I change the way I read the material.</p> <p>Before I study new course material thoroughly, I often skim it to see how it is organized.</p> <p>I ask myself questions to make sure I understand the material I have been studying in this class.</p> <p>I try to change the way I study in order to fit the course requirements and the instructor's</p>	☐	☐	☐	☐	☐	☐	☐
---	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------

teaching style.							
I often find that I have been reading for this class but don't know what it was all about.	☐	☐	☐	☐	☐	☐	☐
I try to think through a topic and decide what I am supposed to learn from it rather than just reading it over when studying for this course.	☐	☐	☐	☐	☐	☐	☐
When studying for this course I try to determine which concepts I don't understand well.	☐	☐	☐	☐	☐	☐	☐
When I study for this class, I set goals for myself in order to direct my activities in each study period.	☐	☐	☐	☐	☐	☐	☐
If I get confused taking notes in class, I	☐	☐	☐	☐	☐	☐	☐

make sure I sort it out afterwards							
--	--	--	--	--	--	--	--

After hitting the submit button, the students are immediately transferred to the VARK questionnaire web site.

Appendix D: VARK Questionnaire

How Do I Learn Best?

VARK Questionnaire version 7.1

Choose the answer which best explains your preference and click the box next to it. Please click more than one if a single answer does not match your perception. Leave blank any question that does not apply.

1. You are about to purchase a digital camera or mobile phone. Other than price, what would most influence your decision?
 - ☐ Trying or testing it.
 - ☐ It is a modern design and looks good.
 - ☐ Reading the details or checking its features online.
 - ☐ The salesperson telling me about its features.
2. You have finished a competition or test and would like some feedback. You would like to have feedback:
 - ☐ using graphs showing what you had achieved.
 - ☐ using examples from what you have done.
 - ☐ from somebody who talks it through with you.
 - ☐ using a written description of your results.
3. You are going to choose food at a restaurant or cafe. You would:
 - ☐ choose from the descriptions in the menu.
 - ☐ look at what others are eating or look at pictures of each dish.
 - ☐ listen to the waiter or ask friends to recommend choices.
 - ☐ choose something that you have had there before.
4. Remember a time when you learned how to do something new. Avoid choosing a physical skill, e.g. riding a bike. You learned best by:
 - ☐ listening to somebody explaining it and asking questions.
 - ☐ diagrams, maps, and charts - visual clues.
 - ☐ written instructions – e.g. a manual or book.
 - ☐ watching a demonstration.
5. I like websites that have:
 - ☐ audio channels where I can hear music, radio programs or interviews.
 - ☐ interesting written descriptions, lists and explanations.
 - ☐ things I can click on, shift or try.

- ☐ interesting design and visual features.
- 6. Do you prefer a teacher or a presenter who uses:
 - ☐ handouts, books, or readings.
 - ☐ diagrams, charts or graphs.
 - ☐ demonstrations, models or practical sessions.
 - ☐ question and answer, talk, group discussion, or guest speakers.
- 7. A website has a video showing how to make a special graph. There is a person speaking, some lists and words describing what to do and some diagrams. You would learn most from:
 - ☐ reading the words.
 - ☐ listening.
 - ☐ watching the actions.
 - ☐ seeing the diagrams.
- 8. Other than price, what would most influence your decision to buy a new non-fiction book?
 - ☐ A friend talks about it and recommends it.
 - ☐ Quickly reading parts of it.
 - ☐ The way it looks is appealing.
 - ☐ It has real-life stories, experiences and examples.
- 9. You have a problem with your heart. You would prefer that the doctor:
 - ☐ showed you a diagram of what was wrong.
 - ☐ used a plastic model to show what was wrong.
 - ☐ gave you something to read to explain what was wrong.
 - ☐ described what was wrong.
- 10. You have to make an important speech at a conference or special occasion. You would:
 - ☐ make diagrams or get graphs to help explain things.
 - ☐ write out your speech and learn from reading it over several times.
 - ☐ write a few key words and practice saying your speech over and over.
 - ☐ gather many examples and stories to make the talk real and practical.
- 11. You are helping someone who wants to go to your airport, the center of town or railway station. You would:
 - ☐ go with her.
 - ☐ write down the directions.
 - ☐ draw, or show her a map, or give her a map.

- ☐ tell her the directions.
12. You are using a book, CD or website to learn how to take photos with your new digital camera. You would like to have:
- ☐ clear written instructions with lists and bullet points about what to do.
 - ☐ a chance to ask questions and talk about the camera and its features.
 - ☐ many examples of good and poor photos and how to improve them.
 - ☐ diagrams showing the camera and what each part does.
13. You want to learn a new program, skill or game on a computer. You would:
- ☐ use the controls or keyboard.
 - ☐ talk with people who know about the program.
 - ☐ follow the diagrams in the book that came with it.
 - ☐ read the written instructions that came with the program.
14. A group of tourists wants to learn about the parks or wildlife reserves in your area. You would:
- ☐ talk about, or arrange a talk for them about parks or wildlife reserves.
 - ☐ give them a book or pamphlets about the parks or wildlife reserves.
 - ☐ show them maps and internet pictures.
 - ☐ take them to a park or wildlife reserve and walk with them.
15. You are planning a vacation for a group. You want some feedback from them about the plan. You would:
- ☐ phone, text or email them.
 - ☐ give them a copy of the printed itinerary.
 - ☐ use a map to show them the places.
 - ☐ describe some of the highlights they will experience.
16. You are going to cook something as a special treat. You would:
- ☐ cook something you know without the need for instructions.
 - ☐ ask friends for suggestions.
 - ☐ use a cookbook where you know there is a good recipe.
 - ☐ look on the Internet or in some cookbooks for ideas from the pictures.

Appendix E: Assignments and Rubrics

Summary

Assessment	Description	Points	Number	Total Points
Assignment 1	Introduction to C#	50	1	50
Assignment 2	Processing data	50	1	50
Assignment 3	Making decisions	50	1	50
Assignment 4	Loops	50	1	50
Assignment 5	Random numbers/Load event	50	1	50
Assignment 6	Methods	50	1	50
Assignment 7	Arrays	50	1	50
Assignment 8	Arrays, Lists	50	1	50
Assignment 9	More processing data	50	1	50
Assignment 10	Classes/Multiform	50	1	50
Examinations	Debugging Exams	50	2	100
	Programming Exams	100	2	200
Daily Quizzes / In-Class Assignments	Quizzes over the Reading Material and Lectures	10	20	200
				1,000

Assignment 8 Description

INTERFACE:

Application Requirements:

- Create a form that when a button is clicked
 - Select and sort 30 random letters in alphabetical order
 - Sort the same 30 random letters in descending order
 - List the 30 **unique** letters in alphabetical order
 - List the 30 **unique** letters in descending alphabetical order
- All display boxes should be read only
- Runs with no errors or warnings
- Include comments
 - Heading comment (see example below) at the top of program

```
// Your Name
// IST 1551
// Assignment #
/* explanation of what the
 * program does
 */
```

- Comment variables (i.e. variables used to collect input, variables used in calculations, putting input into variables, methods, etc.)
- Comment calculation sections
- Comment putting data into variables from form and vice versa
- Comment all functions and methods

Extra Credit:

Be sure focus is always on the button.

Submitting:

Zip the folder that contains all of your files for this assignment and upload to the assignment section in BlackBoard before due date and time.

Name your file YourLastName_YourFirstName.zip

Notes:

Be sure to contact me if you have any questions.

Assignment 8 Rubric

Name Assignment 8 Rubric

Description

Rubric Detail

Criteria	Levels of Achievement		
	Novice	Competent	Proficient
Zipped file is lastName_firstName	0 Points Zipped file is not named lastName_firstName	0 Points n/a	1 Points Zipped file is named lastName_firstName
All objects are properly named	0 Points More than half the objects have improper names	0.5 Points At least half the objects but not ALL the objects are properly named	1 Points All the objects are properly named
All objects with text have appropriate text	0 Points More than half the objects with text have improper text	0.5 Points At least half but not all the objects with text have proper text	1 Points All of the objects with text have appropriate text
There are 4 read only boxes to display letters	0 Points There are less than half of the display areas needed	0.5 Points There are more than half of the display areas needed but not all	1 Points There are 4 read only areas to display the letters
No Errors or warnings before running code	0 Points There are errors before running the code	5 Points There are no errors but there are warnings before running the code	10 Points There are no errors or warnings before running the code
Comments: Header comment (with	0 Points	2 Points	4 Points

Criteria	Levels of Achievement		
	Novice	Competent	Proficient
student's name and a short description of the program), comment before each method/event handler, comments throughout method/event handlers	Missing over half of the comments	At least half of the comments exist but not all of them	All of the comments exist
When the button is clicked, 30 random letters are created	0 Points When the button is clicked, random letters are not created	1 Points When the button is clicked, more than 15, but less than 30, random letters are created	3 Points When the button is clicked, 30 random letters are created
A method exists and works properly to create a random numbers	0 Points A method doesn't exist	1 Points A method exists but doesn't work properly to create random numbers	2 Points A method exists and works properly to create random numbers
The random letters are stored in a List of char type	0 Points The random letters are not stored in a List of char type	0 Points n/a	5 Points The random letters are stored in a List of char type
The random letters are displayed to the form	0 Points The random letters are not displayed to the form	0 Points n/a	3 Points The random letters are displayed to the form
LINQ is used to sort the letters in the char List in alphabetical order. The letters are displayed to the form	0 Points LINQ is not used to sort the letters in the char List in alphabetical order or the letters are not displayed to the form	1 Points Letters are not in the correct order but are displayed to the form	3 Points LINQ is used to sort the letters in the char List in alphabetical order

Criteria	Levels of Achievement		
	Novice	Competent	Proficient
LINQ is used to sort the letters in the char List in reverse alphabetical order. Sorted letters are displayed to the form	0 Points LINQ is not used to sort the letters in the char List in reverse alphabetical order or are not displayed on the form	1 Points Letters are not in the correct order but are displayed to the form	3 Points LINQ is used to sort the letters in the char List in reverse alphabetical order. The letters are displayed to the form
LINQ is used to sort the letters in the char List in alphabetical order and duplicates are removed. Sorted letters are displayed on the form and there are not any duplicates	0 Points LINQ is not used to sort the letters and/or the char List is in alphabetical order and duplicate letters are not removed	1 Points Letters are not in order and/or duplicates are not removed but letters are displayed to the form	3 Points LINQ is used to sort the letters in the char List in alphabetical order and remove the duplicates. Sorted letters are displayed on the form and there are not any duplicates
No errors while running code	0 Points There were errors while running the code	5 Points There are errors when running the code but are handled and the program does not crash	10 Points There were no errors while running the code
EXTRA CREDIT: LINQ is used to sort the letters in the char List in reverse alphabetical order and duplicates are removed. Sorted letters are displayed on the form and there are not any duplicates	0 Points n/a	0 Points n/a	0 Points +5: LINQ is used to sort the letters in the char List in reverse alphabetical order and duplicates are removed. Sorted letters are displayed on the form and there are not any duplicates
View Associated Items			
<div> Print Close Window </div>			

Assignment 8 Lecture Slides

1

2

3

4

5

6

7

8

9

10

11

12

Microsoft University of Science and Technology

Querying an Array of Employee Objects Using LINQ (Cont.)

- The select clause can create a new object of anonymous type (a type with no name), which the compiler generates for you based on the properties listed in the curly braces { }.
- By default, the name of the property being selected is used as the property's name in the result.
- You can specify a different name for the property inside the anonymous type definition.

25

Microsoft University of Science and Technology

Querying an Array of Employee Objects Using LINQ (Cont.)

- The preceding query is an example of a projection—it performs a transformation on the data.
- The transformation creates new objects containing only the `FirstName` and `Last` properties.
- Transformations can also manipulate the data.
 - For example, you could give all employees a 10% raise by multiplying their `MonthlySalary` properties by 1.1.

26

Microsoft University of Science and Technology

Querying an Array of Employee Objects Using LINQ (Cont.)

- Implicitly typed local variables allow you to use anonymous types because you do not have to explicitly state the type when declaring such variables.
- When the compiler creates an anonymous type, it automatically generates a `ToString` method that returns a string representation of the object.

27

Microsoft University of Science and Technology

Introduction to Collections

- The .NET Framework Class Library provides several classes, called *collections*, used to store groups of related objects.
- These classes provide efficient methods that organize, store and retrieve your data *without* requiring knowledge of how the data is being stored.
- The collection class `List<T>` (from namespace `System.Collections.Generic`) does not need to be reallocated to change its size.

28

Microsoft University of Science and Technology

Introduction to Collections (Cont.)

- `List<T>` is called a generic class because it can be used with any type of object.
- `T` is a placeholder for the type of the objects stored in the list.
- Figure 9.5 shows some common methods and properties of class `List<T>`.

29

Microsoft University of Science and Technology

Methods/Properties of Lists

Method or property	Description
<code>Add</code>	Add an element to the end of the list.
<code>Capacity</code>	Property that gets or sets the number of elements a list can store without resizing.
<code>Clear</code>	Removes all the elements from the list.
<code>Contains</code>	Returns true if the list contains the specified element and false otherwise.
<code>Count</code>	Property that returns the number of elements stored in the list.
<code>IndexOf</code>	Returns the index of the first occurrence of the specified value in the list.
<code>Insert</code>	Inserts an element at the specified index.
<code>Remove</code>	Removes the first occurrence of the specified value.
<code>RemoveAt</code>	Removes the element at the specified index.
<code>RemoveAtRange</code>	Removes a specified number of elements starting at a specified index.
<code>Reverse</code>	Reverses the list.
<code>Sort</code>	Sorts the list.
<code>TrimExcess</code>	Reduces the capacity of the list to the number of elements the list currently contains (if any).

Fig. 9.5 | Some methods and properties of class `List<T>`.

30

Microsoft University of Science and Technology

```

1 // Fig. 9.6: ListCollection.cs
2 // Generic List<T> collection demonstration
3 using System;
4 using System.Collections.Generic;
5
6 public class ListCollection
7 {
8     // create a new list of strings
9     // create a new list of strings
10    List<string> list = new List<string>();
11
12    // add "red" to the list
13    list.Add("red"); // append an item to the list
14
15    // insert "yellow" at index 0
16    list.Insert(0, "yellow"); // insert the value at index 0
17
18    // display the colors in the list
19    Console.WriteLine("List contents with counter-controlled loop:");
20    for (int i = 0; i < list.Count; i++)
21        Console.WriteLine("{0}", list[i]);
22 }
23
24 // Fig. 9.6 | Generic List<T> collection demonstration. (Part 1 of 4.)
    
```

31

Microsoft University of Science and Technology

```

23 // display colors using foreach
24 Console.WriteLine("List contents with foreach statement:");
25 foreach (var item in list)
26     Console.WriteLine("{0}", item);
27
28 // add "green" to the end of the list
29 list.Add("green"); // add "green" to the end of the list
30
31 // add "yellow" to the end of the list
32 list.Add("yellow"); // add "yellow" to the end of the list
33
34 // display the list
35 Console.WriteLine("List with two new elements:");
36 foreach (var item in list)
37     Console.WriteLine("{0}", item);
38
39 // remove the first "yellow"
40 list.Remove("yellow"); // remove the first "yellow"
41
42 // display the list
43 Console.WriteLine("Remove first instance of yellow:");
44 foreach (var item in list)
45     Console.WriteLine("{0}", item);
46
47 // remove from index 3
48 list.RemoveAt(3); // remove from index 3
49
50 // Fig. 9.6 | Generic List<T> collection demonstration. (Part 2 of 4.)
    
```

32

Microsoft University of Science and Technology

```

45 // display the list
46 Console.WriteLine("Remove second list element (green):");
47 foreach (var item in list)
48     Console.WriteLine("{0}", item);
49
50 // check if a value is in the list
51 Console.WriteLine("Is 'red' in the list?");
52 Console.WriteLine("Is 'red' in the list?");
53
54 // display number of elements in the list
55 Console.WriteLine("Count: {0}", list.Count);
56
57 // display the capacity of the list
58 Console.WriteLine("Capacity: {0}", list.Capacity);
59
60 // end main
61 // Fig. 9.6 | Generic List<T> collection demonstration. (Part 3 of 4.)
    
```

33

Microsoft University of Science and Technology

```

// Fig. 9.6 | Generic List<T> collection demonstration. (Part 4 of 4.)
// Display list contents with counter-controlled loop: yellow red
// Display list contents with foreach statement: yellow red
// List with two new elements: yellow red green yellow
// Remove first instance of yellow: red green yellow
// Remove second list element (green): red yellow
// Is 'red' in the list?
// Count: 2
// Capacity: 4
    
```

34

Microsoft University of Science and Technology

Introduction to Collections (Cont.)

- The `Add` method *appends* its argument to the end of the `List`.
- The `Insert` method inserts a new element at the specified position.
 - The first argument is an index—as with arrays, collection indices start at zero.
 - The second argument is the value that's to be inserted at the specified index.
 - The indices of elements at the specified index and above increase by one.

35

Microsoft University of Science and Technology

Introduction to Collections (Cont.)

- The `Count` property returns the number of elements currently in the `List`.
- `Lists` can be indexed like arrays by placing the index in square brackets after the `List` variable's name.
- The `Remove` method is used to remove the *first* instance of an element with a specific value.
 - If no such element is in the `List`, `Remove` does nothing.
- `RemoveAt` removes the element at the specified index; the indices of all elements above that index decrease by one.

36

Microsoft University of Science for Engineers

Introduction to Collections (Cont.)

- The `Contains` method returns `true` if the element is found in the `List`, and `false` otherwise.
- `Contains` compares its argument to each element of the `List` in order, so using `Contains` on a large `List` is inefficient.
- The `Capacity` property indicates how many items the `List` can hold without having to grow.
- When the `List` grows, it must create a larger internal array and copy each element to the new array.
- A `List` grows only when an element is added and there is no space for the new element.

37

Microsoft University of Science for Engineers

Querying a Generic Collection Using LINQ

- You can use LINQ to Objects to query `Lists` just as arrays.
- In Fig. 9.7, a `List` of strings is converted to uppercase and searched for those that begin with "R".

38

Microsoft University of Science for Engineers

```

1 // Fig. 9.7: LINQ to Objects using a List of strings
2 // LINQ to Objects using a List of strings
3 using System;
4 using System.Linq;
5 using System.Collections.Generic;
6 public class LINQtoObjectsGeneric
7 {
8     public static void Main( string[] args )
9     {
10         // declare a list of strings
11         List<string> items = new List<string>()
12         items.Add( "Robert" ); // add "Robert" to the end of the list
13         items.Add( "Ruth" ); // add "Ruth" to the end of the list
14         items.Add( "Rita" ); // add "Rita" to the end of the list
15         items.Add( "Rita" ); // add "Rita" to the end of the list
16         items.Add( "Rita" ); // add "Rita" to the end of the list
17
18         // convert all strings to uppercase; select those starting with "R"
19         var startswithR =
20             from item in items
21             let uppercaseItem = item.ToUpper()
22             where uppercaseItem.StartsWith( "R" )
23             select uppercaseItem;
24     }
25 }

```

Fig. 9.7 | LINQ to Objects using a List of strings. (Part 1 of 2.)

39

Microsoft University of Science for Engineers

```

25 // display query results
26 foreach ( var item in startswithR )
27     Console.WriteLine( "ID: ", item );
28
29 Console.WriteLine(); // output end of line
30
31 items.Add( "Robert" ); // add "Robert" to the end of the list
32 items.Add( "Ruth" ); // add "Ruth" to the end of the list
33
34 // display updated query results
35 foreach ( var item in startswithR )
36     Console.WriteLine( "ID: ", item );
37
38 Console.WriteLine(); // output end of line
39
40 // end Main
41 } // end class LINQtoObjectsGeneric

```

RED RUBY RUBY

Fig. 9.7 | LINQ to Objects using a List of strings. (Part 2 of 2.)

40

Microsoft University of Science for Engineers

Querying a Generic Collection Using LINQ (Cont.)

- LINQ's `let` clause can be used to create a new range variable to store a temporary result for use later in the LINQ query.
- The string method `ToUpper` converts a string to uppercase.
- The string method `StartsWith` performs a case sensitive comparison to determine whether a string starts with the string received as an argument.

41

Microsoft University of Science for Engineers

Querying a Generic Collection Using LINQ (Cont.)

- LINQ uses deferred execution—the query executes *only* when you access the results, *not* when you define the query.
- LINQ extension methods `ToArray` and `ToList` immediately execute the query on which they are called.
 - These methods execute the query only once, improving efficiency.

42

Assignment 8 Flow Chart

Assignment 8 Text

List and LINQ

- Although commonly used, arrays have limited capabilities.
- A List is similar to an array but provides additional functionality, such as dynamic resizing.
- A language called SQL is the international standard used to perform queries (i.e., to request information that satisfies given criteria) and to manipulate data.
- C#'s new LINQ (Language-Integrated Query) capabilities allow you to write query expressions that retrieve information from a *variety* of data sources, not just databases.
- LINQ to Objects can be used to filter arrays and Lists, selecting elements that satisfy a set of conditions
- A LINQ provider is a set of classes that implement LINQ operations and enable programs to interact with *data sources* to perform tasks such as projecting, *sorting*, *grouping* and *filtering* elements.

Querying an Array of int Values Using LINQ

- The following example demonstrates querying an array of integers using LINQ
 - Need to add a Using statement
 - Using System.Linq;

Querying an Array Using LINQ (Cont.)

- Repetition statements that filter arrays focus on the steps required to get the results. This is called imperative programming.
- LINQ queries, however, specify the conditions that selected elements must satisfy. This is known as declarative programming.
- The System.Linq namespace contains the LINQ to Objects provider.
- You can declare a local variable and let the compiler infer the variable's type based on the variable's initializer. The var keyword is used in place of the variable's type when declaring the variable.
- A LINQ query begins with a from clause, which specifies a range variable (value) and the data source to query (values).
 - The range variable represents each item in the data source, much like the control variable in a foreach statement.
- If the condition in the where clause evaluates to true, the element is selected.
- A predicate is an expression that takes an element of a collection and returns true or false by testing a condition on that element.
- The select clause determines what value appears in the results.
- The orderby clause sorts the query results in ascending order.

- The descending modifier in the orderby clause sorts the results in descending order.
- Any value that can be compared with other values of the same type may be used with the orderby clause.
- With LINQ, the code that selects elements and the code that displays them are kept separate, making the code easier to understand and maintain.
- A where clause can access the properties of the range variable.
- The conditional AND (&&) operator can be used to combine conditions.
- An orderby clause can sort the results according to multiple properties, specified in a comma-separated list.
- The query results Any method returns true if there is at least one element, and false if there are no elements.
- The query results First method returns the first element in the result.
- The Count method of the query result returns the number of elements in the results.
- The select clause can be used to select a member of the range variable rather than the range variable itself.
- The Distinct method removes duplicate elements, causing all elements in the result to be unique.
- You can specify a different name for the property inside the anonymous type definition.

Introduction to Collections (Cont.)

- The Add method *appends* its argument to the end of the List.
- The Insert method inserts a new element at the specified position.
 - The first argument is an index—as with arrays, collection indices start at zero.
 - The second argument is the value that's to be inserted at the specified index.
 - The indices of elements at the specified index and above increase by one.
- The Count property returns the number of elements currently in the List.
- Lists can be indexed like arrays by placing the index in square brackets after the List variable's name.
- The Remove method is used to remove the *first* instance of an element with a specific value.
 - If no such element is in the List, Remove does nothing.
- RemoveAt removes the element at the specified index; the indices of all elements above that index decrease by one.
- The Contains method returns true if the element is found in the List, and false otherwise.
- Contains compares its argument to each element of the List in order, so using Contains on a large List is inefficient.

- The Capacity property indicates how many items the List can hold without having to grow.
- When the List grows, it must create a larger internal array and copy each element to the new array.
- A List grows only when an element is added and there is no space for the new element.

Querying a Generic Collection Using LINQ

- You can use LINQ to Objects to query Lists just as arrays.
- A List of strings is converted to uppercase and searched for those that begin with "R".
- LINQ's let clause can be used to create a new range variable to store a temporary result for use later in the LINQ query.
- The string method ToUpper converts a string to uppercase.
- The string method StartsWith performs a case sensitive comparison to determine whether a string starts with the string received as an argument.
- LINQ uses deferred execution—the query executes *only* when you access the results, *not* when you define the query.
- LINQ extension methods ToArray and ToList immediately execute the query on which they are called.
 - These methods execute the query only once, improving efficiency.

Assignment 8 Audio Transcription

Today we are going to talk about lists and LINQ – L I N Q in C sharp. Although commonly used, arrays have their limited capabilities. In a list, similar to an array, but it provides more functionality, such as dynamically resizing. There is a language called S Q L, that is the international standard used to perform queries, to request information in data, and to manipulate data. C sharp's new LINQ, Language INtegrated Query, and LINQ is spelled L I N Q. The LINQ capabilities allow you to write query expressions that treat information from a variety of sources and not just databases. LINQ objects can be used to filter arrays and lists and select elements that satisfy a set of conditions. A LINQ provider is a set of classes that implement LINQ operations and enable programs, to interact with data sources to perform tasks such as projecting, sorting, grouping, and filtering elements.

To use LINQ you need to add a using statement at the top of your code. So we will add using system dot L I N Q. Repetition statements that filter arrays focus on the steps required to get the results. This is called imperative programming. LINQ queries, however, specify the conditions that selected elements must satisfy. This is known as declarative programming. The system dot LINQ name space contains a LINQ to the objects provider. You can declare a local variable and a local compiler, infer the variables type based on the variables initializer.

The var keyword is used in place of the variables type when declaring the variable. A LINQ query begins with a from clause, specifies a range variable, and a data source to the query. The range variable represents each item in the data source much like the control variable in the for each statement.

If the condition in the where clause evaluates to true, the element is selected. A predicate is an expression that takes an element of a collection and returns true or false by testing a condition on that element.

The select clause determines what value appears in the results. The order by clause sorts the query results in ascending order.

The descending modifier, in the order by clause, sorts the results in descending order instead of ascending order. Any value, that can be compared with other values of the same type, may be used with the order by clause.

With LINQ, the code that selects elements and the code that displays them are kept separate, making the code easier to understand and maintain. A where clause can access properties of the range variable.

A conditional and, which as you remember, is two ampersands, operator can be used to combine conditions. An order by clause can sort the results according to multiple properties specified in a comma separated list. The query results any method returns true

if there is at least one element and false if there are no elements. The query results first method, returns the first element in the results. The count method in the query result returns the number of elements in the results.

The select clause can be used to select a member of the range variable rather than the range variable itself. The distinct method removes duplicate elements, causing all elements in the result to be unique. You can specify a different name for the property inside the anonymous type definition.

When using a list, the add method appends its arguments to the end of the list. The insert method, inserts the element at the specified position. So, the first argument is an index, as with arrays, collections and lists indices start at zero. The second argument is the value to be inserted at the specified index. The indices at the element at the specified index and above increase by one.

The count property returns the number of elements currently in the list. Lists can be indexed like arrays by placing the index in square brackets after the list's variable name.

The remove method is used to remove the first instance of an element with a specific value. If no such element is in the list, remove does nothing. Remove at removes an element with a specific index. The indices above that element decrease by one.

The contains method, returns true if the element is found in the list and false otherwise. Contains compares its argument to each element in the list, in order, so using contains in a large list is inefficient. The capacity property contains how many items the list can grow without having to grow.

When a list grows, it must create a larger internal array and copy each of the new elements to the array. The list grows, only when an element is added and there is no space for the new element. When you are querying a generic collection, using LINQ, um, you can also access those collections in a list, just like arrays.

LINQ let clause can be used to create a new range variable, to store a temporary result for use later in the LINQ query.

The string method to upper converts the string to upper case. The string method, starts with, prefers a case sensitive comparison, to determine whether a string starts with the string used as an argument. LINQ uses deferred execution.

The query executes, only when you access the results. Not when you define the query. LINQ extension methods to array and to list, immediately execute the query on which they are called. These methods execute the query only once for efficiency.

Assignment 8 Example Program

```
/*Carla Bates
 * IST 1551
 * Assignment 8 in-class
 * This takes a sentence that the user enters and sorts the words from the sentence in
alphabetical order
 *
 * (C) Copyright 2016 by Carla Bates. All Rights Reserved.
 */
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace sentenceWord
{
    public partial class alphaFRM : Form
    {
        public alphaFRM()
        {
            InitializeComponent();
        }

        // button click event handler
        private void alphaBTN_Click(object sender, EventArgs e)
        {
            try // try to catch errors
            {
                // get sentence from text box
                string sentence = sentenceTXT.Text;
                // put words from sentence into string array
                string[] words = sentence.Split();
                // LINQ to sort words in alphabetic order
                var sortedWords =
                    from currentWord in words
                    let lowerCaseWord = currentWord.ToLower()
                    orderby lowerCaseWord descending
                    select lowerCaseWord;
            }
        }
    }
}
```

```

        // loop through sortedWords string array to display them to the form
        foreach ( string thisWord in sortedWords.Distinct() )
        {
            // put each word into box on form
            alphaTXT.Text += thisWord + " ";
        }
    }
    catch (Exception errors) // catch errors
    {
        // message box to display error message
        MessageBox.Show("Errors: " + errors.Message);
    }
}
// text change event handler
private void sentenceTXT_TextChanged(object sender, EventArgs e)
{
    // clear results box when starting a new sentence
    alphaTXT.Clear();
}
}
}

```

Assignment 8 Program Solution

```
/*Carla Bates
 * IST 1551
 * Assignment 8
 * Creates a list of random letters. Then displays the items as
 * 1. Use LINQ to sort the List in ascending order
 * 2. Use LINQ to sort the list in descending order
 * 3. Use LINQ to sort the list in ascending order and removing duplicates
 * 4. Use LINQ to sort the list in descending order and removing duplicates
 *
 * (C) Copyright 2016 by Carla Bates. All Rights Reserved.*/
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace randomLetters
{
    public partial class randomLettersFRM : Form
    {
        public randomLettersFRM()
        {
            InitializeComponent();

            // create random object
            Random randomIndex = new Random();
            //create array of characters
            char[] letters = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k',
'1', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'};
            // create list object to store random letters
            List<char> letterList = new List<char>();

            // method to generate random int that will be the index of the random
letter
            public char getRandomLetter()
            {
                // int variable to get random number
                int thisIndex = randomIndex.Next(26);
                // use random number as index to get random letter from array
                char randomLetter = letters[thisIndex];
                // return random letter
                return randomLetter;
            }

            // submit button click event handler
            private void submitBTN_Click(object sender, EventArgs e)
            {
                // clear result boxes
                firstListTXT.Clear();
            }
        }
    }
}
```

```

        secondListTXT.Clear();
        thirdListTXT.Clear();
        fourthListTXT.Clear();
        letterList.Clear();
// for loop to fill list with random letters
for (int counter = 1; counter <= 30; counter++)
{
    // retrieve random letter from method
    char currentLetter = getRandomLetter();
    // add random letter to list
    letterList.Add(currentLetter);
}

// LINQ query to sort letters in alphabetical order
var lettersAscending =
    from currentLetter in letterList
    orderby currentLetter
    select currentLetter;

// loop through previous results
foreach (char eachLetter in lettersAscending )
{
    // display letters to form
    firstListTXT.Text += eachLetter + " ";
}

// LINQ query to sort letters in descending order
var lettersDecending =
    from currentLetter in letterList
    orderby currentLetter descending
    select currentLetter;

// loop through previous results
foreach (char eachLetter in lettersDecending)
{
    // display letters to screen
    secondListTXT.Text += eachLetter + " ";
}

// loop through alphabetic list and remove duplicates
foreach (var eachLetter in lettersAscending.Distinct())
{
    // display them to form
    thirdListTXT.Text += eachLetter + " ";
}

// loop through descending alphabetic list and remove duplicates
foreach (var eachLetter in lettersDecending.Distinct())
{
    // display them to form
    fourthListTXT.Text += eachLetter + " ";
}
}
}
}

```

Assignment 9
INTERFACE:

Application Requirements:

- Create a form that allows a user to enter a sentence. That sentence is then translated to Pig Latin
 - To translate to Pig Latin, move the first letter of the word to the end then add ay to the end
 - Create a method to translate a word and return the translated word
 - Split the sentence by spaces using the Split method
 - Each word will go into an array
 - Go through the array and send each word to a method to be translated
 - Add each word to the text box to display the results
- Result text box should be read only
- After clicking Translate button, entry text box should clear
- Include comments
 - Heading comment (see example below) at the top of program

```
// Your Name  
// IST 1551
```

```
// Assignment #  
/* explanation of what the  
 * program does  
 */
```

- Comment variables (i.e. variables used to collect input, variables used in calculations, putting input into variables, methods, etc.)
- Comment calculation sections
- Comment putting data into variables from form and vice versa
- Comment all functions and methods

Extra Credit:

Include a Clear (clears text boxes on form) and Exit button (exits application)

Submitting:

Zip the folder that contains all of your files for this assignment and upload to the assignment section in BlackBoard before due date and time.

Name your file YourLastName_YourFirstName.zip

Notes:

Be sure to contact me if you have any questions.

An example that I created is above. Please remember, this is an example and your program does not have to look exactly like this.

For detailed grading information, see the assignment rubric attached to this assignment. *5 extra credit points are available.*

In ALL assignments, all control items should have appropriate labels and names (not checkBox1, checkBox2, label1, label2, form1, etc.).

Assignment 9 Rubric

Name	Assignment 9 Rubric		
Description			
Rubric Detail			
	Levels of Achievement		
Criteria	Novice	Competent	Proficient
Zipped file is lastName_firstName	0 Points Zipped file is not named lastName_firstName	0 Points n/a	1 Points Zipped file is named lastName_firstName
All objects are properly named	0 Points More than half the objects have improper names	0.5 Points At least half the objects but not ALL the objects are properly named	1 Points All the objects are properly named
All objects with text have appropriate text	0 Points More than half the objects with text have improper text	0.5 Points At least half but not all the objects with text have proper text	1 Points All of the objects with text have appropriate text
No errors or warnings before running code	0 Points There are errors before running the code	5 Points There are no errors but there are warnings before running the code	10 Points There are no errors or warnings before running the code
Comments: Header comment (with student's name and a short description of the program), comment before each method/event handler, comments throughout method/event handlers	0 Points Missing over half of the comments	2 Points At least half of the comments exist but not all of them	4 Points All of the comments exist

Criteria	Levels of Achievement		
	Novice	Competent	Proficient
The form includes a text box for the user to enter a sentence and a button to click for the translation to happen	0 Points The form doesn't include a text box for the user to enter a sentence and a button to click for the translation to happen	2 Points The form does not include either a text box or button	5 Points The form includes a text box for the user to enter a sentence and a button to click for the translation to happen
When a button is clicked, the sentence is translated to pig Latin and the sentence is removed from the input box	0 Points When a button is clicked, less than half of the sentence is translated properly	2 Points When a button is clicked, at least half but not all of the sentence is translated properly or the input box is not cleared	5 Points When a button is clicked the sentence is translated to pig Latin correctly
A method exists to translate each word in the sentence to pig Latin	0 Points A method doesn't exist	2 Points A method exists but doesn't translate a word at a time or it doesn't work properly	5 Points A method exists to translate each word in the sentence to pig Latin
Words are separated out from the sentence	0 Points Words are not separated out from the sentence	0 Points n/a	5 Points Words are separated out from the sentence
A read-only text box is used to display all of the translated sentences to the form	0 Points A read-only text box is not used to display all of the translated sentences to the form and some of the previously translated sentences	1 Points A read-only text box is not used to display all of the translated sentences to the form or some of the previously translated sentences	3 Points A read-only text box is used to display all of the translated sentences to the form

Levels of Achievement			
Criteria	Novice	Competent	Proficient
No errors while running code	0 Points There were errors while running the code	5 Points There are errors when running the code but are handled and the program does not crash	10 Points There were no errors while running the code
EXTRA CREDIT: Add clear and exit buttons. Clear should clear the form. Exit button should exit the application.	0 Points There aren't clear and/or exit buttons and/or they do not work properly.	0 Points +2: There is either just a clear or exit button or only 1 button works properly	0 Points +5: Both clear and exit buttons exist and work properly

[View Associated Items](#)

Print

Close Window

Assignment 9 Lecture Slides

1

Assignment 9 Strings and Characters

2

Fundamentals of Characters and Strings

- Characters are the fundamental building blocks of C# source code.
- Every program is composed of characters that, when grouped together meaningfully, create a sequence that the compiler interprets as instructions describing how to accomplish a task.

3

Fundamentals of Characters and Strings

- A program also can contain character constants.
- A character constant is a character that's represented as an integer value, called a *character code*.
 - For example, the integer value 122 corresponds to the character constant 'z'.
 - The integer value 10 corresponds to the newline character '\n'.

4

Fundamentals of Characters and Strings (cont.)

- Character constants are established according to the Unicode character set, an international character set that contains many more symbols and letters than does the ASCII character set.
- A string is a series of characters treated as a unit. These characters can be uppercase letters, lowercase letters, digits and various special characters: +, -, *, /, \$ and others.
- A string is an object of class `string` in the `System` namespace.

5

Fundamentals of Characters and Strings (cont.)

- We write string literals, also called string constants, as sequences of characters in double quotation marks, as follows:
"John Q. Doe"
"9999 Main Street"
"Waltham, Massachusetts"
"(201) 555-1212"

6

Fundamentals of Characters and Strings (cont.)

- A declaration can assign a string literal to a string reference. The declaration `string color = "blue";`
- initializes the string `color` to refer to the string literal object "blue".

7

Fundamentals of Characters and Strings (cont.)

- For example, consider the string "C:\MyFolder\MySubFolder\MyFile.txt" with the following assignment:
`string file = "C:\MyFolder\MySubFolder\MyFile.txt";`
- Using the *verbatim string syntax*, the assignment can be altered to:
`string file = @"C:\MyFolder\MySubFolder\MyFile.txt";`

8

Verbatim Strings

- On occasion, a string will contain multiple backslash characters (this often occurs in the name of a file).
- To avoid excessive backslash characters, it's possible to exclude escape sequences and interpret all the characters in a string literally, using the `@` character to create what's known as a *verbatim string*.
- Backslashes within the double quotation marks following the `@` character are not considered escape sequences.

9

string Constructors

- Class `string` provides eight constructors for initializing strings in various ways.
- The following example demonstrates three of the constructors.

10

```
1 // Fig. 16.1: StringConstructor.cs
2 // Demonstrating string class constructors.
3 using System;
4
5 class StringConstructor
6 {
7     public static void Main(string[] args)
8     {
9         // string constructor
10         string s1 = "C:\\MyFolder\\MySubFolder\\MyFile.txt";
11         string s2 = @"C:\MyFolder\MySubFolder\MyFile.txt";
12         string s3 = new string('x', 10);
13         string s4 = new string("characters", 4, 2);
14         string s5 = new string("characters", 0, 10);
15
16         Console.WriteLine("string s1 = " + s1 + " string s2 = " + s2);
17         Console.WriteLine("string s3 = " + s3 + " string s4 = " + s4);
18         Console.WriteLine("string s5 = " + s5 + " string s6 = " + s6);
19     }
20 }
21 // end Main
22 // end class StringConstructor
23
24 Fig. 16.1 | string constructors. (Part 1 of 2)
```

11

string Indexer, Length Property and CopyTo Method

- The app in Fig. 16.2 presents the *string indexer*, which facilitates the retrieval of any character in the `string`, and the *string property Length*, which returns the length of the `string`.
- The `string` method `CopyTo` copies a specified number of characters from a `string` into a `char` array.

12

Microsoft University of Science and Technology

```

1 // Fig. 16.2 | StringIndexof, Length property and CopyTo method
2 // using the Indexof, property Length and method CopyTo
3 // of class String
4 using System;
5
6 class StringIndexof
7 {
8     public static void Main(string[] args)
9     {
10         string str1 = "Hello there";
11         char[] charArray = new char[5];
12
13         // copy characters from str1 into charArray
14         str1.CopyTo(0, charArray, 0, charArray.Length);
15         Console.WriteLine("The character array is: ");
16
17         for (int i = 0; i < charArray.Length; i++)
18             Console.WriteLine(charArray[i]);
19     }
20 }
21
22 // test Length property
23 Console.WriteLine("Length of str1: " + str1.Length);
24
25 // loop through characters in str1 and display reversed
26 Console.WriteLine("The string reversed is: ");
27
28 for (int i = str1.Length - 1; i >= 0; i--)
29     Console.WriteLine(str1[i]);
30
31 Fig. 16.2 | StringIndexof, Length property and CopyTo method. (Part 1 of 2)

```

13

Microsoft University of Science and Technology

```

32
33 // copy characters from str1 into charArray
34 string.CopyTo(0, charArray, 0, charArray.Length);
35 Console.WriteLine("The character array is: ");
36
37 for (int i = 0; i < charArray.Length; i++)
38     Console.WriteLine(charArray[i]);
39
40 Console.WriteLine("The string reversed is: ");
41
42 for (int i = str1.Length - 1; i >= 0; i--)
43     Console.WriteLine(str1[i]);
44
45 Fig. 16.2 | StringIndexof, Length property and CopyTo method. (Part 2 of 2)

```

14

Microsoft University of Science and Technology

Comparing strings

Comparing Strings with Equals, CompareTo and the Equality Operator (==)

- Class string provides several ways to compare strings.

15

Microsoft University of Science and Technology

Locating Characters and Substrings in strings

- In many apps, it's necessary to search for a character or set of characters in a string.
- For example, a programmer creating a word processor would want to provide capabilities for searching through documents.

16

Microsoft University of Science and Technology

Locating Characters and Substrings in strings

- The app in demonstrates some of the many versions of string methods IndexOf, IndexOfAny, LastIndexOf and LastIndexOfAny, which search for a specified character or substring in a string.

17

Microsoft University of Science and Technology

Extracting Substrings from strings

- Class string provides two Substring methods, which create a new string by copying part of an existing string.

```

string hello = "Hello there";

hello.Substring(6);
hello.Substring(6, 4);

```

18

Microsoft University of Science and Technology

Concatenating strings

- The + operator is not the only way to perform string concatenation.
- The static method Concat of class string concatenates two strings and returns a new string containing the combined characters from both original strings.

19

Microsoft University of Science and Technology

Miscellaneous string Methods

- Class string provides several methods that return modified copies of strings.

```

string hello = "Hello";
string there = "there";

string.Concat(hello, there);

```

20

Microsoft University of Science and Technology

Class StringBuilder

- The string class provides many capabilities for processing strings.
- However a string's contents can never change.
- Operations that seem to concatenate strings are in fact creating new strings—the += operator creates a new string and assigns its reference to the variable on the left of the += operator.
- Every StringBuilder can store a certain number of characters that's specified by its capacity.
- Exceeding the capacity of a StringBuilder causes the capacity to expand to accommodate the additional characters.

21

Microsoft University of Science and Technology

Class StringBuilder (cont.)

- As we'll see, members of class StringBuilder, such as methods Append and AppendFormat, can be used for concatenation like the operators + and += for class string.
- StringBuilder is particularly useful for manipulating in place a large number of strings, as it's much more efficient than creating individual immutable strings.
- Class StringBuilder provides six overloaded constructors.

22

Microsoft University of Science and Technology

Length and Capacity Properties, EnsureCapacity Method and Index of Class StringBuilder

- Class StringBuilder provides the Length and Capacity properties to return the number of characters currently in a StringBuilder and the number of characters that a StringBuilder can store without allocating more memory, respectively.
- These properties also can increase or decrease the length or the capacity of the StringBuilder. Method EnsureCapacity allows you to reduce the number of times that a StringBuilder's capacity must be increased.

23

Microsoft University of Science and Technology

Append and AppendFormat Methods of Class StringBuilder

- Class StringBuilder provides 19 overloaded Append methods that allow various types of values to be added to the end of a StringBuilder.
- The Framework Class Library provides versions for each of the simple types and for character arrays, strings and objects.
- Each method takes an argument, converts it to a string and appends it to the StringBuilder.

24

Microsoft University of Science and Technology

Insert, Remove and Replace Methods of Class StringBuilder (cont.)

- Class StringBuilder also provides method Remove for deleting any portion of a StringBuilder.
- Method Remove takes two arguments—the index at which to begin deletion and the number of characters to delete.
- The sum of the starting index and the number of characters to be deleted must always be less than the length of the StringBuilder; otherwise, the program throws an ArgumentOutOfRangeException.

25

Microsoft University of Science and Technology

Insert, Remove and Replace Methods of Class StringBuilder (cont.)

- Another useful method included with StringBuilder is Replace.
- Replace searches for a specified string or character and substitutes another string or character in its place.

26

Microsoft University of Science and Technology

Char Methods

- C# provides a concept called a struct (short for "structure") that's similar to a class.
- Although structs and classes are comparable, structs represent value types. Like classes, structs can have methods and properties, and can use the access modifiers public and private. Also, struct members are accessed via the member access operator (.).
- All struct types derive from class ValueType, which derives from object.

27

Assignment 9 Flow Chart

Assignment 9 Text

Fundamentals of Characters and Strings

- Characters are the fundamental building blocks of C# source code.
- Every program is composed of characters that, when grouped together meaningfully, create a sequence that the compiler interprets as instructions describing how to accomplish a task.
- A string is a series of characters treated as a unit. These characters can be uppercase letters, lowercase letters, digits and various special characters: +, -, *, /, \$ and others
- A string is an object of class string in the System namespace
- We write string literals, also called string constants, as sequences of characters in double quotation marks, as follows:

"John Q. Doe"

"9999 Main Street"

"Waltham, Massachusetts"

"(201) 555-1212"

- A declaration can assign a string literal to a string reference. The declaration **string** color = **"blue"**;
- initializes the string color to refer to the string literal object "blue".
- For example, consider the string "C:\MyFolder\MySubFolder\MyFile.txt" with the following assignment: **string** file = **"C:\\MyFolder\\MySubFolder\\MyFile.txt"**;

Verbatim Strings

- Using the *verbatim string syntax*, the assignment can be altered to **string** file = **@ "C:\MyFolder\MySubFolder\MyFile.txt"**;
- On occasion, a string will contain multiple *backslash characters* (this often occurs in the name of a file).
- To avoid excessive backslash characters, it's possible to exclude escape sequences and interpret all the characters in a string literally, using the @ character to create what's known as a verbatim string.
- Backslashes within the double quotation marks following the @ character are not considered escape sequences.

Strings

Have their own methods and properties

char

- Data type
- Holds one character
- Enclosed in single quote marks
- Use ToString() to convert to string

Accessing Individual Characters

- Can access letters in a string like array
 - Individually
 - stringVariable[indexOfCharacter]
 - But only for reading
 - Looping
 - for
 - foreach (because a string is a collection)

```
string course = "IST1551";
```

```
char oneLetter;
```

```
oneLetter = course[3];
```

What is the value of oneLetter? 1

for loop

```
string course = "IST1551";
```

```
char oneLetter;
```

```
for(int counter=0; counter<course.Length; counter++)
```

```
{
```

```
    oneLetter = course[counter];
```

```
    MessageBox.Show("Letter "+counter+1 + " is "+oneLetter);
```



```
}
```

foreach loop

```
string course = "IST1551";

foreach(char letter in course)

{

    MessageBox.Show("The next letter is "+ letter);

}
```

Methods for char

- Always return true or false
- char.IsDigit(charVariable)
- char.IsDigit(string, index)
- char.IsLetter(charVariable)
- char.IsLetter(string, index)
- char.IsLetterOrDigit(charVariable)
- char.IsLetterOrDigit(string, index)
- char.IsLower(charVariable)
- char.IsLower(string, index)
- char.IsUpper(charVariable)
- char.IsUpper(string, index)
- char.ToUpper(charVariable)
- char.ToLower(string[0])
- char.IsPunctuation(charVariable)
 - !"#\$%&^()*,-.:;?@[_{} }
- char.IsPunctuation(string, index)
- char.IsWhiteSpace(charVariable)
 - Space, tab, linefeed, carriage-return, formfeed, vertical-tab, and newline
- char.IsWhiteSpace(string, index)

Example

```
• char.IsDigit(charVariable)
string course = "IST1551";

if (char.IsDigit(course[2]))
```



```

{
    MessageBox.Show("The third character is a number");
}

else

{
    MessageBox.Show("The third character is not a number");
}

```

```

    • char.IsDigit(string, index)
string course = "IST1551";

if (char.IsDigit(course, 2))

{
    MessageBox.Show("The third character is a number");
}

else

{
    MessageBox.Show("The third character is not a number");
}

```

Case Conversion

```

    • char.ToUpper
    • char.ToLower
string firstName, lastName;

firstName = firstNameTXT.Text;

lastName = lastNameTXT.Text;

```

```
char letterFirst = char.ToUpper(firstName[0]);
```

```
char letterLast = char.ToUpper(lastName[0]);
```

substring

- `stringVar.Contains(substring)`
- `stringVar.StartsWith(substring)`
- `stringVar.EndsWith(substring)`
- `stringVar.Substring(start)`
- `stringVar.Substring(start, count)`
- `stringVar.IndexOf(substring)`
- `stringVar.IndexOf(substring, start)`
- `stringVar.IndexOf(substring, start, count)`
- `stringVar.IndexOf(char)`
- `stringVar.IndexOf(char, start)`
- `stringVar.IndexOf(char, start, count)`

Examples:

```
    • course.Contains("1551")
string course = "IST1551";

string course2 = "IST1552";

if (course.Contains("51"))
{
    MessageBox.Show("Contains 51");
}

if (course2.Contains("51"))
{
    MessageBox.Show("Contains 51");
}
```

- `stringVar.IndexOf(substring, start, count)`

- Returns -1 if not found

```
string dessert = "tutti frutti ice cream";  
  
int position = dessert.IndexOf("ti", 4, 3);  
  
What value is in position variable? -1  
  
int position = dessert.IndexOf("ti", 4, 10);  
  
Now? 10
```

Substring

- `stringVar.Substring(start)`
 - Returns characters of `stringVar` beginning at `start` index and going to the end of the string
- `stringVar.Substring(start, count)`
 - Returns count number of characters from `stringVar` beginning at `start` index

- `stringVar.Substring(start, count)`
- Returns -1 if not found

```
string dessert = "tutti frutti ice cream";  
  
string parts = dessert.Substring(4, 3);  
  
The value of parts is "i f"
```

Methods for Modifying Strings

- Does not change the actual string variable but the changed string is stored in a second variable
- Insert
 - Inserts a string into another string
 - `stringVar.Insert(start, stringToInsert);`
- Remove
 - Removed specified characters from a string
 - `stringVar.Remove(start);`
 - `stringVar.Remove(start, count);`
- ToLower

- Converts entire string to lowercase
- ToUpper
 - Converts entire string to uppercase
- Trim
 - Removes all spaces at beginning and end
- TrimStart
 - Removes all spaces at the beginning
- TrimEnd
 - Removes all spaces at the end

Tokenizing strings

- Breaking a string into “tokens”
- String with a series of words or items separated by the same character
- Character is called a delimiter
- If entered by user
 - Trim string
 - Get rid of extra spaces

Split Method

- Used to tokenize a string
- Items from string are returned in an array of strings
- Passing null as argument uses white-space characters as delimiter
- Pass a char array as delimiter
 - Each element in array is used as delimiter

Example

```
string sentence = "I like programming";

string[] allWords = sentence.Split();

foreach (string eachWord in allWords)
{
    MessageBox.Show("This word "+ eachWord);
}

string colors= "blue,red,green";
```

```
char[] comma = {','};  
  
string[] allColors = colors.Split(comma);  
  
foreach (string eachColor in allColors)  
{  
  
    MessageBox.Show("This color"+ eachColor);  
  
}
```

Join Method

- Concatenates items from a string array to a string variable

```
string[] allColors = {"blue","red","green"};  
  
string colors = String.Join(",", allColors);
```

Assignment 9 Audio Transcript

Okay, so, chapter 9 we are looking at characters and strings. So, you are familiar with strings. We put double quotes around strings when we set a variable equal to them. When we create a string variable. Sometimes, you might need to do, like a path name, so like c colon backslash my folder backslash IST 1551. So, anyway, those backslashes are special characters in c sharp. So, what you can do is put ampersand in front of the quote marks before you create your string literal and that is called verbatim string syntax. And, then, that tells c sharp to look at that as an actual backslash and not use it as the special meaning that it has in c sharp.

Strings have their own special properties. We also have a char data type. Char data type only holds one character and it is enclosed in single quote marks. If you need to change a char variable to a string, then you can do the dot to string open and closed parentheses at the end of the char variable. You can access individual characters in a string just like you can in an array. So, you would, individually you would just have your string variable name and then the index of the character you want to access inside square brackets just se do for arrays. But that is just accessing, not actually change the string variable but that way you can read from it.

You can also loop through your string variables to get each um character in the loop if you needed to. Um, a for or foreach. Remember a foreach is great to use with a collection. A string is just a collection of characters.

Then we have methods for char variables and they return true or false. So you will have char dot is digit. And the i in is is capitalized and the d in digit is capitalized. And then you would put your char variable in parentheses. And then you put your char variable in parentheses. You can either put your char variable in parentheses there or you can put a string and then do comma and enter an index. So it would sti ll be a char data type because you are only getting one character from the string and a char is one character. So, not only do we have is digit, we have is letter, we have is letter or digit. So, a letter or a number. Um, is lower, checks to see if it is lowercase. Is upper, checks to see if it is uppercase. And then we have a to upper and to lower also. Where, you can um set a variable equal to that, that, and change the case uppercase or lowercase but you will have to set that equal to a variable. It doesn't actually change, um, the string that you are looking at.

Then we have is punctuation. Punctuation is an exclamation point, quote sign, pound sign, percent sign, ampersand, forward slash, backward slash, open parenthesis, close parenthesis, asterisk, comma, dash, period, colon, semicolon, question mark, at sign, square bracket open, square bracket close, underline, curly bracket open and curly bracket close. Those are all the punctuations. And then we also have is white space. So, true false, is it a white space. Conversion, so if you want to convert something, we have the char dot to upper and char dot to lower, but you have to set the variable equal to that

and then put what you are changing in the parentheses. Like we said before, either a char variable or um a string and an index.

With substring we have several methods, too. We have, and with substring, you always put the string name variable first and then dot and whichever substring method you want. So we have contains, that checks to see, and then in parentheses you put what you are looking for. Um, you have the string variable dot starts with, and in parentheses you put what you are looking for and it will see if it is at the start. String var dot ends with substring and in the parentheses. And then check for ending with that. Um, string variable dot substring start will give you part of the string and it will start at a certain place and go to the end of the string. Or you can use the string variable dot substring start comma count and the parentheses. And start comma count that is in the parentheses, will tell it where to start and how many characters to get. String var dot index of substring. So it's going to look for the substring that you have in parentheses in the string variable that is mentioned before right before the index of and it is going to give you the index of where that substring starts. And you can do it with either a substring or char. And, you can tell it where to start and give it a count, so how many of those. Or tell it where to start and it will go from the start to the end of the string.

Okay. So, we also have methods for modifying strings. It doesn't actually change the string. You have to have a string variable equal to what you are modifying and it will go, be stored in that second variable. So, we have string variable dot insert. And then in parentheses is start comma string to insert. So it's going to insert where you tell it to start and it's going to insert the string, um, that you put in that as you put as the second parameter in that string variable dot insert. Then, we also have string variable dot remove. And, you tell it were to start or you have string variable dot remove where you tell it start and how many, um, characters to delete. We also have to lower, which converts to lowercase. To upper which converts to upper case. Trim which removes all the spaces from the beginning and the end. Trim start, which removes all the spaces at the beginning. Trim end, which removes all the spaces at the end.

So, breaking a string into pieces is called tokens, tokenizing strings. And then, so if you are separating by a certain character, that character is called a delimiter. And, so we have the split method, which we had in the last assignment, or assignment 8. So, if you don't put anything in the parentheses, after a split, then it's automatically going to split on spaces. And, if you do want to split on something besides spaces, then you have to define those characters to a char array and use that char array inside the parentheses, then it will break on different things than spaces.

And, then we have the join method, which concatenates items from a string array to a string variable. So, we have the string variable defined, with everything separated by commas or whatever you want. And then you have the string, with that, of your new variable and you'll do string dot join, and then in parentheses, you will pass it two parameters, the first one is going to be, um, what is separating the, um, what you want the different items from the array separated by, or your delimiter. And, so, if you use like a

comma, you will put that in quote marks. And, then you do comma and give it the name of the array that you want to put in the string.

And, that's all for the audio this week. Good luck!

Assignment 9 Example Program

```
/*Carla Bates
 * IST 1551
 * Assignment 9 in-class
 * This takes a sentence that the user enters and switches the first and last letter of each
word
 * and display the sentence back
 *
 * (C) Copyright 2016 by Carla Bates. All Rights Reserved.
 */
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace switchLetters
{
    public partial class switchFRM : Form
    {
        public switchFRM()
        {
            InitializeComponent();
        }

        // method to switch the first and last letter of each word
        private string switchFirstLast(string currentWord)
        {
            string newWord = ""; // variable to hold new word

            try // catch errors
            {
                char firstLetter = currentWord[0]; // grab the first letter
                char lastLetter = currentWord[currentWord.Length - 1]; // grab the last letter
                currentWord = currentWord.Remove(0, 1); // remove the first letter
                currentWord = currentWord.Remove(currentWord.Length - 1); // remove the
last letter
                newWord = lastLetter.ToString() + currentWord + firstLetter.ToString(); //
create the new word
            }
            catch { }
        }
    }
}
```

```

    }
    catch(Exception error) // catch errors
    {
        MessageBox.Show("Error: " + error.Message); // display error
    }

    return newWord; // return new word
}

// button click event handler
private void switchBTN_Click(object sender, EventArgs e)
{
    try // catch errors
    {
        string sentence = sentenceTXT.Text; // grab sentence from text box
        string[] allWords = sentence.Split(); // put words into an array

        string newSentence = ""; // variable to hold new sentence
        foreach (string word in allWords) // loop through each word in the array
        {
            string newWord; // variable for new word
            newWord = switchFirstLast(word); // call method to switch letters and get
new word
            newSentence += newWord + " "; // add new word to new sentence variable
        }

        switchedTXT.Text = newSentence; // display new sentence
    }
    catch(Exception error) // catch errors
    {
        MessageBox.Show("Error in button click event handler" + error.Message); //
display errors
    }
}

// clear button click
private void clearBTN_Click(object sender, EventArgs e)
{
    try // catch errors
    {
        // clear boxes
        sentenceTXT.Clear();
        switchedTXT.Clear();
    }
    catch (Exception error) // catch errors

```

```

    {
        MessageBox.Show("Error at clear: " + error.Message); // display errors
    }
}

// exit button click event handler
private void exitBTN_Click(object sender, EventArgs e)
{
    try // catch errors
    {
        this.Close(); // close application
    }
    catch (Exception error) // catch errors
    {
        MessageBox.Show("Error at exit: " + error.Message); // display errors
    }
}
}
}

```

Assignment 9 Program Solution

```
/*Carla Bates
 * IST 1551
 * Assignment 9
 * This takes a sentence that the user enters and translates it to pig latin
 * then displays the new sentence on the form
 *
 * (C) Copyright 2016 by Carla Bates. All Rights Reserved.
 */
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace pigLatin
{
    public partial class pigLatinFRM : Form
    {
        public pigLatinFRM()
        {
            InitializeComponent();

            // method to translate each word to Pig Latin
            private string pigLatin(string currentWord)
            {
                string newWord = ""; // variable to hold new word

                try // catch errors
                {
                    char firstLetter = currentWord[0]; // grab the first letter
                    currentWord = currentWord.Remove(0, 1); // remove the first letter
                    newWord = currentWord + firstLetter.ToString() + "ay"; // create
the new word

                }
                catch (Exception error) // catch errors
                {
                    MessageBox.Show("Error in translator method: " + error.Message); //
display error
                }
                return newWord; // return new word
            }

            // translate button click event handler
            private void translateBTN_Click(object sender, EventArgs e)
            {
                string originalSentence = sentenceTXT.Text; //grab sentence from form
                string[] allWords = originalSentence.Split(); // put words into an
array
            }
        }
    }
}
```

```

        string newSentence = ""; // variable to hold new sentence
        foreach (string word in allWords) // loop through each word in the
array
        {
            string newWord; // variable for new word
            newWord = pigLatin(word); // call method to translate word to pig
latin and get new word
            newSentence += newWord + " "; // add new word to new sentence
variable
        }

        sentenceTXT.Clear();
        translatedTXT.Text += newSentence + "\r\n"; // display new sentence
    }

    // clear button click event handler
    private void clearBTN_Click(object sender, EventArgs e)
    {
        try // catch errors
        {
            // clear boxes
            sentenceTXT.Clear();
            translatedTXT.Clear();
        }
        catch (Exception error) // catch errors
        {
            MessageBox.Show("Error at clear: " + error.Message); // display
errors
        }
    }

    // exit button click event handler
    private void exitBTN_Click(object sender, EventArgs e)
    {
        try // catch errors
        {
            this.Close(); // close application
        }
        catch (Exception error) // catch errors
        {
            MessageBox.Show("Error at exit: " + error.Message); // display
errors
        }
    }
}

```

Appendix F: Post Assignment Questionnaire

Q1 Comment about good and bad points of the flow chart. Did the flow chart help you? If so, how?

Q2 Did you refer to the flow chart while you were working on the assignment?

☐ Yes

☐ No

Q3 Comment about the good and bad points of the audio file. Did the audio file help you? if so, how?

Q4 Did you refer to the audio file while you were working on the assignment?

☐ Yes

☐ No

Q5 Comment about the good and bad points of the text file. Did the text file help you? If so, how?

Q6 Did you refer to the text file while you were working on the assignment?

☐ Yes

☐ No

Q7 If you did not attend class for the in-class assignment, what was your main motivation for not attending class?

Q8 Comment about the good and bad points of the video if you watched it. Did the video help you? If so, how?

Q9 Did you watch the video while working on the assignment?

☐ Yes

☐ No

Q10 If you watched the video more than once, why did you?

Q11 Do you have any comments about attending class for the in-class assignment being optional and posting the video?

Q12 Rank the item helpfulness with 1 being most helpful.

_____ Audio file

_____ Example program

_____ Flow diagram

_____ Text file

Q13 Do you have any comments about the helpfulness of the options?

Q14 Rank your preference of items to use.

- Audio file
- Example program
- Flow diagram
- Text file

Q15 Do you have any comments about the preference of the options?

Appendix H: Screen Shots Showing the Order of Teaching Mode

Assignment 8

Build Content ▾ Assessments ▾ Tools ▾ Partner Content ▾

Assignment 8
Enabled: Statistics Tracking
Attached Files: Assignment8description.doc (180 KB)
[See description](#)

Assignment8: Flowchart
Enabled: Statistics Tracking
Attached Files: Assignment 8 Flowchart.pdf (240.537 KB)

You should be able to open the file. You may have to "Allow Blocked Content". Let me know if you have problems.
If you cannot view the .vsdx file, try opening the PDF file instead.

Assignment 8: Text Information
Enabled: Statistics Tracking
Attached Files: textinfo.docx (21.347 KB)

Read this for 10 points to count toward your in-class assignment and quizzes. Submit a comment "Read this" or anything else.

Assignment8: Audio
Enabled: Statistics Tracking
Attached Files: Assignment8.mp3 (11.911 MB)

Assignment 9

Build Content ▾ Assessments ▾ Tools ▾ Partner Content ▾

Assignment 9
Enabled: Statistics Tracking
Attached Files: Assignment9description.doc (138.5 KB)

Assignment9: Audio File
Enabled: Statistics Tracking
Attached Files: Assignment9audio.mp3 (9.575 MB)

Listen to the audio file, post a comment, and submit for 10 poing in-class assignment/quiz score

Assignment9: Flowchart
Enabled: Statistics Tracking
Attached Files: Assignment 9 Flowchart.pdf (151.132 KB)

After viewing, submit a comment "I viewed the flowchart" for your 10 points to be included with your in-class assignments/quizzes.

Assignment9: Text Information
Enabled: Statistics Tracking
Attached Files: TextDocument.docx (23.801 KB)

After reading, submit a comment "I read the document" for you 10 points to be included with your in-class assignments/quizzes.

Appendix I: Tegrity (Example Assignment) Video

Appendix J: VARK® Survey Results

The VARK Questionnaire

Your VARK Results

Your scores were:

Visual 2

Aural 3

Read/Write 5

Kinesthetic 5

You have a multimodal learning preference. (RK)

Use the following page for study strategies that apply to your learning preference:

[Multimodal strategies](#)

[Read/Write Strategies](#)

[Kinesthetic Strategies](#)

To interpret your results try a [Personalised VARK profile](#) - approx. US\$15 for several pages of helpful suggestions for learning.

Personal Learning Profile

Your personalised VARK Profile provides more information about your learning preferences. The VARK questionnaire provides four scores and there are a very large number of combinations of those scores. You can order a VARK Profile - a report based on your scores from the questionnaire. Every attempt is made to personalize the Profile so that it describes strategies suited to your learning.

[More Information](#)

[Purchase a Short Profile](#)

[Purchase a Full Academic Profile](#)

[Purchase a Full Business Profile](#)

Vita

Since Fall semester of 2014 I have been employed as an Associate Teaching Professor in the Business Information and Technology Department at Missouri University of Science and Technology. Previously, I was employed as a Programmer Analyst in the Information Technology Department for 20 years during which, 10 of those years, I was an Adjunct Instructor for Business Information and Technology. I earned my Masters of Science in Information Science and Learning Technology from the Business Information and Technology Department at Missouri University of Science and Technology in 2005 and my Bachelors of Science in Management Systems from the Business Information and Technology Department at Missouri University of Science and Technology in 1996.