NAMING CONVENTIONS IN OBJECT-ORIENTED PROGRAMMING

Stephen Dane Tilzey

THESIS APPROVED:

Chair, Advisory Committee	Dr. Anitha Chennamaneni Printed Name	04/05/18 Date
Committee Member	Dr. Randall Brown Printed Name	4/5/18 Date
Committee Member	Dr. Ferdi Eruysal Printed Name	4/5/18 Date
Department Chair	Dr. Anitha Chennamaneni Printed Name	04/12/18 Date
College Dean	Dr. Larry Garner Printed Name	4/12/18 Date
Director of Graduate Studies	Dr. Kellie Cude Printed Name	5 June 2018 Date

NAMING CONVENTIONS IN OBJECT-ORIENTED PROGRAMMING

THESIS

Presented to the Office of Graduate Studies
Texas A&M University
Central Texas
In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE INFORMATION SYSTEMS

By

STEPHEN DANE TILZEY

Killeen, Texas
May 2018

Tilzey, Stephen D., *Naming conventions in object-oriented programming,* Master of Science (Information Systems), May, 2018, 79 pp., 5 tables, bibliography, 42 titles.

This study examined the need for a standardized naming convention and the effect that a detailed naming convention, Hungarian Notation, has on the readability and comprehension of source code. Participants of the study were asked to score two sets of source code snippets using a Likert-scale based on its readability, simplicity, and understanding. Results found that there is a current need for a standardized naming convention. The readability and comprehension measures were significantly different between the two code sets indicating they are positively affected by the usage of a detailed naming convention. The results also found that the choice of a programming language had no significant effect on the need for a standard naming convention or the readability and comprehension of the source code. Based on the results, the study's limitations and recommendations are discussed.

TABLE OF CONTENTS

LIST OF GRAPHICS

TABLE

Chapter I

Introduction

Object-oriented programming, developed more than five decades ago (Derk, 2011), is one of the most common and utilized type of coding in our society today. Object-oriented programming operates with variables, known as objects, created using a premade or custom blueprint, classes. One of the advantages of object-oriented programming is to develop your own names for all the variables, classes, methods, functions, and other entities. Though extremely useful, the ability to custom name everything in the source code can cause serious issues and confusion, (Gopstein et al., 2017). In the fifty years that object-oriented programming has existed, there has been no standardized naming convention chosen.

Naming conventions are rules and practices in order to better the readability and understanding of the source code. Despite knowing the benefits and purpose, the American education system teaches that naming conventions be recommended, but not required (Della & Clark, 2000). A programming textbook of the C coding language (Kernighan & Ritchie, 1988) gives only the following reference to naming conventions for programming:

> Names are made up of letters and digits; the first character must be a letter. The underscore ``_" counts as a letter; it is sometimes useful for improving the readability of long variable names. Don't begin variable names with underscore, however, since library routines often use such names. Upper and lower case

> letters are distinct, so x and X are two different names. Traditional C practice is to use lower case for variable names, and all upper case for symbolic constants. (p.35)

The lack of a standardized naming convention can have detrimental effects on the source code. Several studies, (Corbi, 1989; Lientz, Swanson, & Tompkins, 1978), have shown that the maintenance and revision of source code is the most time and budget consuming stage of code development. The reason behind this is that the source code relies heavily on its readability, and is significantly decreased by the lack or improper use of a standard naming convention, (Bacchelli & Bird, 2013).

Literature Review

Over the years, several academic studies have researched and evaluated the effects of the practices of naming conventions and the generalized usage of naming in coding (Sharif & Maletic, 2010; Deissenboeck & Pizka, 2006; Guerrouj, 2013; Kuhn, Ducasse, & Girba , 2007; Schober et al., 2009; and Binkley, Davis, Lawrie, & Morrell, 2009). However, none have studied the same exact research questions of the need for a standardized naming convention and the effect of a detailed naming convention on readability and comprehension. The studies discussed in the following review were obtained through Google Scholar, ACM Digital Library, Academic OneFile, Gale PowerSearch, Texas A&M University Central Texas Library, and the Fiske Public Library of Massachusetts.

Sharif and Maletic (2010) conducted research over identifier naming conventions. The purpose of their research was to conduct an empirical study to determine if an identifier naming convention, such as camelCase and under_score, affect the comprehension of source code. They claimed that since identifiers consist of a majority of the source code, if a certain identifier naming style could increase the speed of comprehension, it would result in an increase of entire program understanding. For their research, they focused on two research questions:

> RQ1: Does identifier style affect the accuracy and time needed to read and detect correct identifiers?
>
> RQ2: Is the visual effort needed to read and detect correct identifiers the same for camel-case and underscore styles? (p. 2)

Using a Tobii 1740 eye tracker, Sharif and Maletic (2010) were able to measure the fixation count of the visual effort needed for the participants to answer the questions correctly. The questions were a series of phrases that the participants would study and then receive four possible answers from which to choose. After conducting their research, they discovered that identifier style significantly influences the time and visual effort needed to detect the correct identifier from the constructed phrase. Their data illustrated that the under_score identifier style was overall quicker by approximately twenty percent compared to the camelCase identifier style. Since the Hungarian Notation naming convention utilizes a more detailed basis of the under_score identifier style, the results of

Sharif and Maletic's study facilitate the hypothesis that a naming convention is more effective than without.

Deissenboeck and Pizka (2006) also contended the significance of identifier naming in programming. In their paper, Deissenboeck and Pizka argued that an identifier naming style in programming would be an opportunity to facilitate the source code comprehension. This in turn would result in an increase to the productivity and quality of the source code maintenance and evolution through its development process. Rather than recommend a standardized naming convention, Deissenboeck and Pizka instead proposed to implement and utilize a tool-supported Identifier Dictionary (IDD). An IDD would work similarly to a Data Dictionary by storing information about the identifiers such as their name, data type of the object, and a detailed description. Though implementing such a tool would significantly increase the overhead for the source code development, their study still illustrates the beneficial effects that an identifier naming convention would have on the source code readability and comprehension.

Guerrouj (2013) found that in order to read and comprehend the program, perform reverse engineering, or complete any re-documentation, one would have to understand the source code. However, Guerrouj (2013) also discovered that approximately more than half of linguistic information in the source code of a program consists of identifiers (e.g., names of classes, methods, parameters, or attributes, etc.). If a large percentage of source code is made of identifiers, it would be an intelligent and advisable decision to ensure the source code is following the most efficient naming convention. In a circumstance such as

this, a standardized detailed identifier naming convention would be the most appropriate and best choice. Kuhn et al. (2007) discovered in their research over semantic clustering that bad naming, such as using too generic variable names, arbitrary and random names, or abbreviations that are too cryptic to be externally understood cause the main threat to external validation of source code. In order to combat against this, they found that the best result was to utilize a good naming convention and have logical and predetermined identifiers. Schober et al. (2009) also discovered in their research that decreases in source code readability was primarily caused by a lack or inconsistent usage of a naming convention. Similarly, Schober et al. (2009) believed that a clear, unambiguous, and univocal general naming convention should be applied to all object-oriented programming.

Binkley et al. (2009) performed an experiment to understand the readability of identifiers in programming. Their central hypothesis considered that the speed and accuracy of manipulating source code in a program was significantly affected by the identifier style. Similar to the Sharif and Maletic (2010) experiment, Binkley et al. (2009) had their participants study a constructed identifier phase and then choose the identifier. The two identifiers chosen for their research was under_score and camelCase. Their research experiment consisted of four hypotheses:

Hypothesis 1:

- H10: Correctness is the same regardless of the Style of the identifier.
- H1A: Correctness is affected by identifier Style.

Hypothesis 2:

- H20: Find Time is the same regardless of the Style of the identifier.
- H2A: Find Time is affected by identifier Style.

Hypothesis 3:

- H30: The effect of Style on Correctness is independent of Training.
- H3A: The effect of Style on Correctness lessens due to Training.

Hypothesis 4:

- H40: The effect of Style on Find Time is independent of Training.
- H4A: The effect of Style on Find Time lessens due to Training. (p. 5)

Binkley et al. (2009) reported their experiment resulted in almost half of those without computer science training preferring under_score. Almost forty percent of those using camelCase with previous computer science training were found to prefer under_score. As the same with Sharif and Maletic (2010) results, the under_score identifier naming convention was the most preferred, indicating that the Hungarian Notation naming convention, which is typically considered a more detailed subset of under_score, would be the most effective identifier naming convention.

As mentioned earlier, Hungarian Notation is the chosen identifier naming convention for this research experiment. However, a slightly modified and modern Hungarian Notation naming convention will be utilized during the experiment. There are two main versions of Hungarian Notation used, Systems and Apps. Systems Hungarian Notations uses a prefix of the variable type such as "int" for integer or "str" for string.

Apps Hungarian Notation uses a prefix to specify the variable's purpose such as "pcv" for private class variable or "lpi" for loop index. This research study utilizes a modern combination of both Systems and Apps Hungarian Notation as the detailed naming convention. Charles Simonyi created the first Hungarian Notation by using prefixes to indicate the format and data type. Since then, Hungarian Notation has evolved throughout many different programming languages and has been modified or adapted to satisfy the needs in the source code.

Depending on the chosen style of Hungarian Notation, it can closely follow under_score or camelCase. This experiment will use the style most closely related to under_score. Hungarian Notation follows some of the same principles as under_score in that the identifiers are separated by a "_" (underscore). The identifiers, which are typically an abbreviation of the data type or format of the named object, are located at the beginning of the name to increase the initial find time and readability. The end of the name holds the actual meaningful name of the named object. An example of this would be naming the local integer variable that is supposed to be holding the value for the salary, "l_int_salary". The "l" represents that it is a local scoped variable. The "int" represents that it is an integer data type.

Hungarian Notation was chosen for this research experiment because there was a gap in the literature of previous studies by failing to address it compared to other naming conventions. Also, personal experience utilizing the detailed Hungarian Notation factored into choosing it for this study.

Several studies have shown that though the camelCase naming convention, which utilizes compound words and phrases, is widely used in programming, it has received much criticism over its decreased readability to the source code due to the compounding of the object name (Sharif & Maletic, 2010; Binkley et al., 2009). Another study's results showed that another naming convention, Snake Case that utilizes full compound words separated by an underscore for each word, has an even decreased readability compared to camelCase (Binkley et al., 2009). The PascalCase naming convention was also not chosen for this research due to it being almost identical to camelCase minus that each word, including the first, is capitalized. It can be assumed that the PascalCase naming convention would have approximately the same decreased readability as camelCase.

In my personal experience of writing computer programs for over seven years, I have utilized many naming conventions, but eventually use and prefer a modern modified version of Hungarian Notation. As an undergraduate computer science student at Texas A&M University - Central Texas (A&M - Central Texas), I spent several semesters as an unofficial non-paid programming tutor. The students I tutored ranged in their background demographics and their level of experience with coding. During this time, I observed that the students who had the more difficult time understanding the code material either did not utilize a naming convention properly or were lacking one entirely. I noticed that after reviewing code that was in Hungarian Notation and advising them to use it, almost each of them noticeably improved their readability and comprehension of their source code. During my tutoring sessions, I questioned a few of the students about their experience and education in naming conventions, and most responded that they were taught that it

was recommended, but not required. Li and Prasad (2005) researched over whether coding standard were being taught effectively in the classroom environment. Their research questions were:

> RQ1: How should we teach coding standards?
>
> RQ2: Is there any common way to implement coding standards in all programming courses?
>
> RQ3: How should we assess the learning of coding standards? (p. 1)

Li and Prasad (2005) collected data from 1st year students and then again two years later, when they were 3rd year students. They discovered through their analysis that their data indicated most students consider coding standards to be an important concept in programming, but tend to not comply with them. They found that there is a need for a more effective teaching strategy for standards in programming courses. Their research data also showed that students within the two-year span reported an increase in the opinion that coding standards: makes code look better, helps in reducing errors, and helps for team communication. In addition, over half of the 1st year level students responded that they knew Hungarian Notation after the enforced courses and lectures.

Chapter II

Goal of Study

Research Questions

The purpose of this study was to conduct research to examine the need for a naming convention standard in object-oriented programming languages, primarily Java, C#, and VB. This research investigated if using a detailed naming convention, Hungarian Notation, is more effective in the readability and comprehension of the source code compared to the same code that does not utilize a naming convention. The following research questions were addressed:

- Question 1: Is there currently a need for a naming convention standard in object-oriented programming?
- Question 2: Does the usage of a detailed naming convention affect the readability of the source code?
- Question 3: Does the usage of a detailed naming convention affect the comprehension of the source code?
- Question 4: Does a choice of programming language affect the need for a naming convention standard in object-oriented programming?
- Question 5: Does a choice of programming language affect the effect that a detailed naming convention has on readability of the source code?
- Question 6: Does a choice of programming language affect the effect that a detailed naming convention has on comprehension of the source code?

Research Hypotheses

Need for a naming convention. Guerrouj's (2013) research found that over fifty percent of all linguistic information within source code consists of identifiers suggesting that naming in programming is a crucial task and should be held to the most efficient level of standard. Li and Prasad (2005) also found in their research that most students consider coding standards to be an important concept in programming, with a majority of students increasing their opinions of coding standard between their first and third years of school. These previous findings support the following hypothesis:

- H_1: There is a need for a standardized naming convention in object-oriented programming.

Readability. Research conducted by Binkley et al. (2009) and Sharif and Maletic (2010) found that the style of identifier used in source code significantly affects the readability speed and accuracy. Schober et al. (2009) also found that source code readability decreased due to a lack of a naming convention. These findings support the following hypothesis:

- H_2: A detailed naming convention affects the readability of source code compared to a non-utilizing naming convention.

Comprehension. Kuhn et al. (2007) found in their research that bad naming or a lack of a proper naming convention was the main threat to external comprehension of source code. Guerrouj (2013) also found that being able to understand the source code

and its associated identifiers was crucial in comprehending the program itself. These findings support the following hypothesis:

- H_3: A detailed naming convention affects the comprehension of source code compared to a non-utilizing naming convention.

Programming Language: The choice of language was examined as a factor on the hypotheses due to their differences. Chen (2010) found in his comparative study of popular programming languages that though similar, C# is more appropriate for application and web application development, Java is more appropriate for mobile and server programming, and Visual Basic is more appropriate for front end programming of databases. These findings support the following hypotheses:

- H_4: The choice of programming language affects the need for a standardized naming convention in object-oriented programming.
- H_5: The choice of programming language affects the detailed naming convention's effect on readability of source code compared to a non-utilizing naming convention.
- H_6: The choice of programming language affects the detailed naming convention's effect on comprehension of source code compared to a non-utilizing naming convention.

Chapter III

Method

Participants

This research study's sample was drawn from A&M - Central Texas students who were currently enrolled in at least one computer science or computer information systems course and were currently over the age of 18 years. The study was approved by A&M - Central Texas Institutional Review Board (IRB) with participation being voluntary and completely confidential, in which no identifying information was collected. Data was collected from forty-nine students ($N = 49$), however, two questionnaires were incomplete and had to be dropped. Therefore, only forty-seven participants ($N = 47$) and their questionnaire responses could be utilized in analysis for the final data set. About 82.98% ($n = 39$) of participants were male. About 21.28% ($n = 10$) of participants were between the18-24 age group, 27.66% ($n = 13$) were between the 25-30 age group, 34.04% ($n = 16$) were between the 31-45 age group, and 17.02% ($n = 8$) were over the age of 45.

When asked to select their most applicable current occupation, 51.06% ($n = 24$) of participants identified as students, 36.17% ($n = 17$) identified as being employed, only 2.13% ($n = 1$) identified as being unemployed, and 10.64% ($n = 5$) identified as being retired.

Of the students who participated in this research study, 10.64% (n = 5) reported high school as their current highest level of education, 40.43% (n = 19) reported an Associate's degree, 38.30% (n = 18) reported a Bachelor's degree, and 10.64% (n = 5) reported a Master's degree. No student who participated selected the Doctor of Philosophy (PhD) option. Though it was optional, of the 89.36% (n = 42) participants who selected a degree option, only 42.86% (n = 18) listed their degree in the given space of the demographic section. 55.56% (n = 10) listed their degree as Computer Information Systems, 22.22% (n = 4) listed Computer Science, 5.56% (n = 1) listed Mathematics, 5.56% (n = 1) listed Management of Information Systems, 5.56% (n = 1) listed Masters of Business Administration, and 5.56% (n = 1) listed Software Engineering & Database Design.

When asked about their programming experience, 53.19% (n = 25) of participants reported less than one-year experience, 36.17% (n = 17) of participants reported between two and three years' experience, 4.26% (n = 2) of participants reported four to five years' experience, and 6.38% (n = 3) reported more than six years of experience. Participants were then asked to select the number of programming languages they felt they knew. Surprisingly, 8.51% (n = 4) of the sample reported they knew zero programming languages. The majority of student participants reported they knew in between one to three languages, 25.53% (n = 12) reported one language, 27.66% (n = 13) reported two languages, and 25.53% (n = 12) reported three languages. The remaining 12.77% (n = 6) reported they knew either four or more programming languages. Students were asked to list their programming languages in the given space of the demographic section. The

frequencies were as follows: Visual Basic was listed 21 times, Java was listed 17 times, C++ was listed 17 times, C# was listed 16 times, Python was listed 10 times, HTML was listed 7 times, C was listed twice, and JavaScript, Objective-C, and PHP were all listed just once.

Demographics Results

Table I: Demographics Results Table

Variable	**Count (%)**
Gender:	
Male	39 (82.98%)
Female	8 (17.02%)
Age:	
<18	0 (0%)
18-24	10 (21.28%)
25-30	13 (27.66%)
31-45	16 (34.04%)
45>	8 (17.02%)
Current Occupation:	
Student	24 (51.06%)
Employed	17 (36.17%)
Un-employed	1 (2.13%)

Retired	5 (10.64%)
Current Education:	
High School	5 (10.64%)
Associates Degree	19 (40.43%)
Bachelor's Degree	18 (38.3%)
Master's Degree	5 (10.64%)
PhD	0 (0%)
Programming Experience:	
<1 year	25 (53.19%)
2 - 3 years	17 (36.17%)
4 - 5 years	2 (4.26%)
>6 years	3 (6.38%)
Programming Languages Known:	
Zero	4 (8.51%)
One	12 (25.53%)
Two	13 (27.66%)
Three	12 (25.53%)
Four or more	6 (12.77%)

Materials

Volunteer lab flyer and email. Volunteer flyers were distributed in the A&M - Central Texas computer lab to recruit possible participants for the study, (please see Appendix A). Two emails were sent to every instructor of a Computer Science or Computer Information System course of the Fall 2017 and Spring 2018 semesters. The first email was intended to be forwarded to the students enrolled in the courses informing them of the research study. The second email to the instructors inquired into arranging a time to conduct a voluntary survey after the class was finished.

Participation consent form. A consent form with all the information regarding the research study was verbally addressed with and distributed to participants before the study (please see Appendix B). The participants and their responses remained anonymous throughout the entire study. Participants were also informed that their participation was entirely voluntary and they would receive no completion incentives from the survey conductor or their instructor if conducted in a class-wide environment.

Questionnaire. A questionnaire was created to investigate the research questions and hypotheses of this research study, (please see Appendix C, D, and E). If a student consented to volunteer in the study, they were given the choice of a questionnaire in one of three popular object-oriented programming languages, C# (please see Appendix C), Java (please see Appendix D), and Visual Basic (please see Appendix E). The first two pages of the questionnaires were arranged in a randomized order. One page consisted of source code snippets utilizing a detailed naming convention, modified Hungarian

Notation, whereas the other page consisted of source code snippets that did not utilize any known naming convention. The third page was a demographic and reflection questionnaire for the participants to fill out after completing the first two pages.

Source code snippet ratings. Utilizing a modified version of the approach taken in the research experiment over code readability conducted by Buse and Weimer (2010), the first section of the questionnaire has the participant rate source code snippets, (please see Appendix C, D, and E). Participants score the source code snippet, typically no more than 6 lines of source code written at an introductory level, between 1 (worst) to 5 (best) in each of the factor categories of readability, simplicity, and understanding. There were two sets of source code snippets created for the study that alternate between the two pages the participants receive in order to eliminate the factor of the student being familiar with the same code twice. However, the source code snippets in each set were identically created in regards to the length, difficulty, and purpose in order to eliminate the factor of one set being easier to read or understand.

Source code snippet questions. In addition to the source code ratings, participants were asked to answer two open-ended response questions reference a source code snippet. The questions were primarily created not to question the overall understanding of the code, but to investigate if the student can determine the answer by the naming convention alone. After writing their responses, each question followed with a yes or no checkbox asking if they found either the usage or the lack of a naming convention beneficial in responding.

Reflection and demographic questionnaire. After completing the first two pages of the questionnaire, the participants would then complete the reflection and demographic sections, (please see Appendix F). The reflection questions section asked the students to score their opinions to three statements using a Likert-scale ranging from 1 (strongly agree) to 7 (strongly disagree). The three statements were as followed:

- There should be a naming convention standard in object-oriented programming.
- It was easier to read the code that did not follow a naming convention.
- It was easier to read the code that followed a naming convention.

The final section of the questionnaire was the demographic questions. These questions asked the participant to self-report their gender, age, current occupation, current highest education optionally listing their majors obtained, programming experience in terms of years, and number of programming languages known optionally listing the languages.

Procedure

This study followed a modified version of the procedure approach taken in the research experiment over code readability conducted by Buse and Weimer (2010). After this research study was reviewed and approved by the A&M - Central Texas' IRB, participants were recruited during the Spring 2018 semester for eight weeks. Several recruitment flyers were distributed from the A&M - Central Texas computer lab with permission from the Academic Technologies department.

The population sample that this research study analyzed consisted primarily of questionnaires that were conducted in a classroom after the class had finished. The survey administer informed the students of the consent form and the purpose, rules, and directions of the research study and its questionnaire. Students were informed that their participation was voluntary and that they would receive no form of incentive from either the survey conductor or their professor. If a student volunteered to participant in the research study, they chose one of the three questionnaire themed programming languages that they felt the most comfortable and knowledgeable with. If any student chose to exit out of the study during the conduction of the questionnaire, their non-completed questionnaire was discarded and they were thanked for their time. Once all questionnaires were passed out, a timer was started to signal to the students to begin answering the form. The students were instructed to complete the questionnaire in the exact order of the pages as they were first given to them. The participants then completed the code snippet rating and question sections for both the non-usage naming convention and detailed naming convention pages. As they completed a page, participants were instructed to write the current elapsed time as indicated on the projected timer. Once the code snippet pages were completed, participants were asked to complete the reflection and demographic sections on the final page. After the participants completed the questionnaire and turned them in, the survey conductor thanked them for their voluntary participation and research opportunity.

Chapter IV

Results

Paired Sample T-Test

The paired sample t-test, also known as the dependent sample t-test, was utilized in this research study to analyze the data provided by the participants. Of the data collected, six pairs of observations were analyzed using the IBM SPSS software provided by A&M - Central Texas. The six pairs were split into utilizing the detailed naming convention and not utilizing a naming convention, and were as follows: readability, simplicity, understanding, total score, number of questions correct, and time taken to complete the survey. Readability, simplicity, and understanding were scored between a lowest possible score of 5 and a highest possible score of 30. Total score was calculated between a lowest possible score of 15 and a highest possible score of 90, as it is the combined score of readability, simplicity, and understanding. Number of questions correct was scored between a lowest possible score of 0 and a highest possible score of 2. Time taken to complete the survey was recorded in total seconds. After initial analysis, the data was further split into three categories based on the programming language taken for the participant's survey; however, the six pairs remained constant for each analysis. For all analyses of the pairs, a statistical significance value of 0.05 and a two-tailed probability equation were utilized.

The t-test found that readability was significantly different between the detailed naming convention ($M = 25.17$, $SD = 4.687$) and non-utilizing naming convention ($M =$

21.81, *SD* = 5.815) groups, *t* = 3.259, *p* = 0.002. Simplicity was found to be significantly different between the detailed naming convention (*M* = 24.36, *SD* = 4.834) and non-utilizing naming convention (*M* = 21.81, *SD* = 5.480) groups, *t* = 2.541, *p* = 0.015. Understanding was found to also be significantly different between the detailed naming convention (*M* = 25.15, *SD* = 5.082) and non-utilizing naming convention (*M* = 20.13, *SD* = 5.889) groups, *t* = 4.859, *p* = <0.001. The total score was significantly different between the detailed naming convention (*M* = 74.68, *SD* = 13.939) and non-utilizing naming convention (*M* = 63.74, *SD* = 16.14) groups, *t* = 3.854, *p* = <0.001. Number of questions right was also found to be significantly different between the detailed naming convention (*M* = 1.36, *SD* = 0.735) and non-utilizing naming convention (*M* = 0.13, *SD* = 0.337) groups, *t* = 11.610, *p* = <0.001. The difference between the two groups on time taken to complete the survey was found to not be significant, however being close to the significance level, *t* = -1.905, *p* = 0.063, with the detailed naming convention (*M* = 229.62, *SD* = 71.543) and non-utilizing naming convention (*M* = 260.89, *SD* = 118.93).

Table II: Paired T-Test Results

	Result	**DNC Mean** (*SD*)	**NNC Mean** (*SD*)	**T-Statistic**	**Probability**
Readability	Significantly Different	**25.17** (4.687)	**21.81** (5.815)	*t* = 3.259	*p* = 0.002
Simplicity	Significantly Different	**24.36** (4.834)	**21.81** (5.480)	*t* = 2.541	*p* = 0.015

Understanding	Significantly Different	**25.15** (5.082)	**20.13** (5.889)	$t = 4.859$	$p = <0.001$
Total Score	Significantly Different	**74.68** (13.939)	**63.74** (16.14)	$t = 3.854$	$p = <0.001$
Questions Right	Significantly Different	**1.36** (0.735)	**0.13** (0.337)	$t =$ 11.610	$p = <0.001$
Completion Time	Not Significantly Different	**229.62** (71.54)	**260.89** (118.93)	$t =$ - 1.905	$p = 0.063$

After splitting the data into only those who received the survey written in C# (n = 11), the paired sample t-test was analyzed again on the six observation pairs. The t-test found that readability was still significantly different between the detailed naming convention (M = 26.82, SD = 2.75) and non-utilizing naming convention (M = 22.82, SD = 5.193) groups, t = 2.544, p = 0.029. Simplicity was found to not be significantly different between the detailed naming convention (M = 25.27, SD = 3.849) and non-utilizing naming convention (M = 24.09, SD = 4.571) groups, t = 0.673, p = 0.516. Understanding remained to be significantly different between the detailed naming convention (M = 26.91, SD = 2.773) and non-utilizing naming convention (M = 22.82, SD = 4.854) groups, t = 2.636, p = 0.025. Total score was not significantly different at the set level, though close, between the detailed naming convention (M = 79, SD = 8.379) and non-utilizing naming convention (M = 69.73, SD = 14.001) groups, t = 2.030, p =

0.07. Number of questions right remained significantly different between the detailed naming convention (M = 1.82, SD = 0.603) and non-utilizing naming convention (M = 0.27, SD = 0.467) groups, t = 7.455, p = <0.001. The difference between the two groups on time taken to complete the survey was found again to not be significant, t = -1.998, p = 0.074, with the detailed naming convention (M = 246.27, SD = 79.243) and non-utilizing naming convention (M = 300.45, SD = 116.848).

When the data was split into only those who received the survey written in Java (n = 16) and analyzed again with the paired sample t-test, four of the six pairs were not significantly different between the detailed naming convention and non-utilizing naming convention groups. Readability became not significantly different between the detailed naming convention (M = 24.63, SD = 5.795) and non-utilizing naming convention (M = 21.94, SD = 5.471) groups, t = 1.513, p = 0.151. Simplicity also became not significantly different between the detailed naming convention (M = 24.13, SD = 5.784) and non-utilizing naming convention (M = 22.06, SD = 5.221) groups, t = 1.126, p = 0.278. Understanding, however, remained significantly different between the detailed naming convention (M = 25.13, SD = 5.852) and non-utilizing naming convention (M = 20.31, SD = 5.134) groups, t = 2.693, p = 0.017. Total score was no longer significantly different between the two groups, t = 1.849, p = 0.084, with the detailed naming convention (M = 73.88, SD = 16.998) and non-utilizing naming convention (M = 64.31, SD = 15.391). Number of questions right remained significantly different between the detailed naming convention (M = 1.19, SD = 0.834) and non-utilizing naming convention (M = 0.19, SD = 0.403) groups, t = 4.899, p = <0.001. The t-test also found that the time

taken to complete the Java survey was not significantly different between the two groups, $t = -1.569$, $p = 0.138$, with the detailed naming convention ($M = 207.63$, $SD = 70.936$) and non-utilizing naming convention ($M = 264.25$, $SD = 160.456$).

The data was lastly split into only those who received the survey written in Visual Basic ($n = 20$) and analyzed again, with most pairs remaining significantly different. Readability, however, did not remain significantly different between the detailed naming convention ($M = 24.7$, $SD = 4.543$) and non-utilizing naming convention ($M = 21.15$, $SD = 6.556$) groups, $t = 1.950$, $p = 0.066$. The t-test found that simplicity was significantly different between the detailed naming convention ($M = 24.05$, $SD = 4.662$) and non-utilizing naming convention ($M = 20.35$, $SD = 5.905$) groups, $t = 2.294$, $p = 0.033$. Understanding remained significantly different between the detailed naming convention ($M = 24.2$, $SD = 5.357$) and non-utilizing naming convention ($M = 18.5$, $SD = 6.613$) groups, $t = 3.133$, $p = 0.005$. Total score also remained significantly different between the two groups, $t = 2.730$, $p = 0.013$, with the detailed naming convention ($M = 72.95$, $SD = 13.816$) and non-utilizing naming convention ($M = 60.0$, $SD = 17.457$). Number of questions right was found to be significantly different between the detailed naming convention ($M = 1.25$, $SD = 0.639$) and non-utilizing naming convention ($M = 0.0$, $SD = 0.0$) groups, $t = 8.753$, $p = <0.001$. The t-test found that the time taken to complete the Visual Basic survey paired differences were almost identical between the detailed naming convention ($M = 238.05$, $SD = 66.87$) and non-utilizing naming convention ($M = 236.45$, $SD = 72.026$) groups, $t = 0.08$, $p = 0.937$.

Analysis of Variance

A one-way analysis of variance, also known as ANOVA, was also utilized in this research study to analyze the mean group variation. The ANOVA test was conducted over three measures from the data provided by the participants: readability, simplicity, and understanding. Readability, simplicity, and understanding were scored between a lowest possible score of 5 and a highest possible score of 30. The ANOVA test split the measures into groups by the programming language the survey was taken in, C#, Java, or Visual Basic, and then compared between the detailed naming convention and non-utilizing naming convention groups. For all analyses of the observed measures, a statistical significance value of 0.05 was utilized.

When analyzing the data by the ANOVA test, it was found that the detailed naming convention readability scores are not significantly different between the C# (M = 26.82, SD = 2.75), Java (M = 24.625, SD = 5.8), and Visual Basic (M = 24.7, SD = 4.543) groups, F = 0.884, p = 0.42. It was found that the non-utilizing naming convention readability scores are also not significantly different between the C# (M = 22.82, SD = 5.193), Java (M = 21.94, SD = 5.471), and Visual Basic (M = 21.15, SD = 6.556) groups, F = 0.289, p = 0.751.

The detailed naming convention simplicity scores were found to not be significantly different between the C# (M = 25.27, SD = 3.849), Java (M = 24.125, SD = 5.783), and Visual Basic (M = 24.05, SD = 4.662) groups, F = 0.248, p = 0.782. The non-utilizing naming convention simplicity scores were also found to not be significantly

different between the C# ($M = 24.09$, $SD = 4.571$), Java ($M = 22.06$, $SD = 5.221$), and Visual Basic ($M = 20.35$, $SD = 5.905$) groups, $F = 1.733$, $p = 0.189$.

The detailed naming convention understanding scores were found to not be significantly different between the C# ($M = 26.91$, $SD = 2.773$), Java ($M = 25.125$, $SD = 5.852$), and Visual Basic ($M = 24.2$, $SD = 5.357$) groups, $F = 1.009$, $p = 0.373$.The non-utilizing naming convention understanding scores were also found to not be significantly different between the C# ($M = 22.82$, $SD = 4.854$), Java ($M = 20.31$, $SD = 5.134$), and Visual Basic ($M = 18.5$, $SD = 6.613$) groups, $F = 2.004$, $p = 0.147$.

Table III: ANOVA Detailed Naming Convention Results

	Result	**C# Mean**	**Java Mean**	**VB Mean**	**F-Statistic**	**Probability**
Readability	Not Significantly Different	**26.82** (2.75)	**24.625** (5.8)	**24.7** (4.543)	$F =$ 0.884	$p = 0.42$
Simplicity	Not Significantly Different	**25.27** (3.849)	**24.125** (5.78)	**24.05** (4.662)	$F =$ 0.248	$p = 0.782$
Understanding	Not Significantly Different	**26.91** (2.773)	**25.125** (5.85)	**24.2** (5.357)	$F =$ 1.009	$p = 0.373$

Table IV: ANOVA Non-Utilizing Naming Convention Results

	Result	**C# Mean**	**Java Mean**	**VB Mean**	**F-Statistic**	**Probability**
Readability	Not Significantly Different	**22.82** (5.193)	**21.94** (5.471)	**21.15** (6.556)	$F =$ 0.289	$p = 0.751$
Simplicity	Not Significantly Different	**24.09** (4.571)	**22.06** (5.221)	**20.35** (5.905)	$F =$ 1.733	$p = 0.189$
Understanding	Not Significantly Different	**22.82** (4.854)	**20.31** (5.134)	**18.5** (6.613)	$F =$ 2.004	$p = 0.147$

Reflection Scores

After completing the code snippet sections of the survey, participants were then asked to score their opinion on three statements. The three statements were as follows:

1. There should be a naming convention standard in object-oriented programming.
2. It was easier to read the code that did not follow a naming convention.
3. It was easier to read the code that followed a naming convention.

Participants could score each statement utilizing the Likert-scale of values between 1 and 7. The numerical values were assigned as 1 being strongly agree with the

statement and 7 being strongly disagree with the statement. A scored value of 4 was neutral towards the statement.

Of the total participants ($N = 47$), 78.72% ($n = 37$) of students reported there should be a naming convention standard in object-oriented programming, 19.15% ($n = 9$) of students reported there should not be a naming convention standard in object-oriented programming, and only 2.13% ($n = 1$) of students reported neutral towards the need for a standardized naming convention. It was found that 14.89% ($n = 7$) of students reported it was easier to read the source code that did not follow a naming convention, 76.60% ($n = 36$) of students reported it was not easier to read the source code that did not follow a naming convention a disagreeable opinion, and 8.51% ($n = 4$) of students reported neutral towards the easiness to read the non-utilizing naming convention source code. It was found that 80.85% ($n = 38$) of students reported it was easier to read the source code that followed a naming convention, 17.02% ($n = 8$) reported it was not easier to read the source code that followed a naming convention, and 2.13% ($n = 1$) of students reported neutral towards the easiness to read the detailed naming convention source code.

When split into only those who received the survey written in C# ($n = 11$), 72.73% ($n = 8$) of students reported there should be a naming convention standard in object-oriented programming, 27.27% ($n = 3$) reported there should not be a naming convention standard in object-oriented programming a disagreeable opinion, and no students reported neutral towards the need for a standardized naming convention. It was found that 9.09% ($n = 1$) reported it was easier to read the source code that did not follow

a naming convention, 81.82% (n = 9) reported it was not easier to read the source code that did not follow a naming convention, and 9.09% (n = 1) reported neutral towards the easiness to read the non-utilizing naming convention source code. It was found that 81.82% (n = 9) reported it was easier to read the source code that did follow a naming convention, 18.18% (n = 2) reported it was not easier to read the source code that did follow a naming convention, and no students reported neutral towards the easiness of reading the detailed naming convention source code.

For those who received the survey written in Java (n = 16), 87.5% (n = 14) of students reporting there should be a naming convention standard in object-oriented programming, 12.5% (n = 2) of students reporting there should not be a naming convention standard in object-oriented programming, and no students reporting neutral towards the need for a standardized naming convention. It was found that 18.75% (n = 3) of students reported it was easier to read the source code that did not follow a naming convention, 75% (n = 12) of students reported it was not easier to read the source code that did not follow a naming convention, and 6.25% (n = 1) of students reported neutral towards the easiness to read the non-utilizing naming convention source code. It was found that 81.25% (n = 13) of students reported it was easier to read the source code that did follow a naming convention, 12.5% (n = 2) of students reporting it was not easier to read the source code that did follow a naming convention a disagreeable opinion, and 6.25% (n = 1) of students reporting neutral towards the easiness to read the detailed naming convention source code.

Of those who received the survey written in Visual Basic (n = 20), 75% (n = 15) of students reported there should be a naming convention standard in object-oriented programming, 20% (n = 4) of students reported there should not be a naming convention standard in object-oriented programming, and 5% (n = 1) of students reported neutral towards the need for a standardized naming convention. It was found that 15% (n = 3) of students reported it was easier to read the source code that did not follow a naming convention, 75% (n = 15) of students reported it was not easier to read the source code that did not follow a naming convention, and 10% (n = 2) of students reported neutral towards the easiness to read the non-utilizing naming convention source code. It was lastly found that 80% (n = 16) of students reported it was easier to read the source code that did follow a naming convention, 20% (n = 4) reported it was not easier to read the source code that did follow a naming convention, and no students reported neutral towards the easiness to read the detailed naming convention source code.

Results Hypotheses

Table V: Results Hypotheses Table

	Mean	**Std. Dev.**	**Statistic Test**	**Probability**	**Result**
H_1	2.3617	2.2109	-	-	Supported
H_2	25.17/ 21.81	4.69/ 5.82	t = 3.259	p = 0.002	Supported
H_3	25.15/ 20.13	5.08/ 5.89	t = 4.859	p = <0.001	Supported
H_4	2.9/ 1.8/ 2.5	2.6/ 1.8/ 2.2	F = 0.864	p = 0.428	Not Supported

H_5	26.8/ 24.6 / 24.7	2.8/ 5.8/ 4.5	$F = 0.884$	$p = 0.42$	Not Supported
H_6	26.9/ 25.1/ 24.2	2.7/ 5.9/ 5.4	$F = 1.009$	$p = 0.373$	Not Supported

***Significance level set to 0.05.**

Chapter V

Discussion

Naming Convention

The present research study examined whether there was a need for a standardized naming convention and if the usage of a detailed naming convention affects the student's readability and understanding of the source code.

In regards to the first research question, it was hypothesized that there is a need for a standardized naming convention in object-oriented programming. Almost eighty percent of the participants reported there is a need for a standardized naming convention. Findings from this research study are in agreement with the findings of Li and Prasad (2005) that most students consider coding standards to be an important programming concept.

In regards to the second research question, it was hypothesized that a detailed naming convention affects the readability of source code compared to a non-utilizing naming convention. Results again support this hypothesis. This study found the readability scores were significantly different between the detailed naming convention and non-utilizing naming convention groups. The detailed naming convention was found to have a positive effect towards the readability. Findings from this research study are in agreement with the findings of Schober et al. (2009) that a lack or inconsistent usage of a naming convention would decrease the source code readability.

In regards to the third research question, it was hypothesized that a detailed naming convention affects the comprehension of source code compared to a non-utilizing naming convention. Results from this study again support this hypothesis. This study found the understanding scores were significantly different between the detailed naming convention and non-utilizing naming convention groups. The detailed naming convention was found to have a positive effect towards the comprehension of the source code. Findings from this research study are in agreement with the findings of Kuhn et al. (2007) that using too generic variable names or arbitrary and random names that are too cryptic are the main threat to the external understanding of source code.

In regards to the fourth research question, it was hypothesized that the choice of a programming language does affect the need for a standardized naming convention in object-oriented programming. The results found that the need for a standardized naming convention was not significantly different between the programming languages examined in this study: C#, Java, and Visual Basic.

In regards to the fifth research question, it was hypothesized that the choice of a programming language does affect the detailed naming convention's effect on readability of the source code. The results found that the readability score was not significantly different between the programming languages examined in this study, thus it does not support the hypothesis that there is an effect on readability.

In regards to the sixth and final research question, it was hypothesized that the choice of a programming language does affect the detailed naming convention's effect on

comprehension of the source code. The results found that the understanding score was not significantly different between the programming languages examined in this study, thus it does not support the hypothesis that there is an effect on comprehension.

Limitations

There were several limitations of this research study that must be considered. The sample for this study consisted only of students from A&M - Central Texas who volunteered to participate. This sample may not be truly representative of the entire university student population of A&M - Central Texas or the national college student population. The sample may not be representative of the non-student population as well. The sample had an uneven gender distribution with most participants being male. However, the computer science field has an uneven male dominated gender distribution to begin with (Cohoon & Aspray, 2006). Only current students of a computer science or computer information systems course that had an on-campus class were recruited for this study. A&M - Central Texas has an average student age that is higher than most traditional universities. A younger student population may not have been represented by the study's older student sample. A&M - Central Texas is also an upper level school. In addition, there was an uneven distribution of grade levels that participated in the study. The majority of participants were undergraduate (Bachelor's level) students, with only one course of graduate (Master's Level) students participating.

Due to the limited time given to conduct the research, only one semester (Spring 2018) worth of volunteers was collected.

A limited sample size was also another limitation of this research study. Although a minimum expected sample size of 20 students was set, 47 students, double the minimum, volunteered to participate in the study. Though higher than the expected sample size, this study's sample size is still considered small.

Another possible limitation to this research study was the design and layout of the survey. After completing the survey, several students voiced their comments that the survey forms, although easy to read, had too many sections to complete. Future researchers would need to consider the possibility of spreading the survey's sections out over more pages or condensing the sections into a more comfortable layout for the participants.

Future Research

In order to determine if a standardized naming convention is needed in both the educational and professional settings, additional future research is required. Future research could determine if teaching a standardized naming convention at the beginning of a student's programming education and experience could factor towards their readability and understanding of the source code. Additional future research could determine if the professional programming experience of an individual is a factor towards his/her opinion on the need for a standardized naming convention. Research towards determining the most efficient naming convention, in terms of readability and understanding, to be implemented as the standard is also necessary. If a standard naming convention is eventually chosen, it would be most effective to teach the naming

convention at the start of a programmer's education (Li & Prasad, 2005). Teachers could implement the standard into their curriculums using the four-step standards procedure as laid out by the United States Army standards application model (Shipley & Burks, 1997). The first step is to "Set the Standard". Once research has statistically determined the most efficient naming convention, it will then be set as the object-oriented programming standard. The second step is to "Communicate the Standard". Teachers would be informed of the naming convention standard. The third step is to "Train the Standard". Teachers would adapt their current curriculums to implement the new set standard for naming conventions to train their students. The fourth and final step is to "Enforce the Standard". Teachers would have to enforce the set naming convention standard in all their students work and lessons in order to facilitate the full benefits of the researched standard.

Conclusion

This research study offered a look into the possibility of a standardized naming convention in object-oriented programming. Almost 78.7% of the student participants reported there was a need for a naming convention standard to be set in object-oriented programming. In regards to determining a naming convention and researching into the differences between a detailed naming convention and non-utilizing naming convention, the results of this study found that the readability and comprehension of source code is significantly different between the two groups, with students scoring the detailed naming convention on average higher by fifteen and twenty-five points, respectfully. The results

also found that the three object-oriented programming languages of C#, Java, and Visual Basic did not appear to have a statistically significant effect on the need for a standardized naming convention or the readability and comprehension of source code. However, further research is required to investigate the possible factors and affected measures not evident in this research due to its limitations. In conclusion, this research study provides empirical evidence that there is a need for a standardized naming convention and that a detailed naming convention does positively affect the readability and understanding measures of source code.

REFERENCES CITED

Bacchelli, A., & Bird, C. (2013, May 18). *Expectations, outcomes, and challenges of modern code review*. Paper presented at the Proceedings of the 2013 International Conference on Software Engineering, San Francisco, CA. Retrieved from https://dl.acm.org/citation.cfm?id=2486882

Binkley, D., Davis, M., Lawrie, D., & Morrell, C. (2009, May 17). *To camelcase or under_score*. Paper presented at the IEEE 17th International Conference on Program Comprehension 2009, Vancouver, Canada. doi:10.1109/icpc.2009.5090039

Buse, R., & Weimer, W. (2010, July). *Learning a metric for code readability*. Paper presented at the IEEE Transactions on Software Engineering, Piscataway, NJ. Retrieved from https://dl.acm.org/citation.cfm?id=1850615

Chen, H. (2010). *Comparative study of C, C++, C#, and Java programming languages* (Thesis). Vaasan Ammattikorkeakoulu, University of Applied Sciences, Vaasa, Finland. Retrieved from https://www.theseus.fi/bitstream/handle/10024/16995/Chen_Hao.pdf?sequence=1

Cohoon, J. M., & Aspray, W. (Eds.) (2006). *Women and information technology: Research on underrepresentation* (Vol. 1). Cambridge, MA: The MIT Press.

Corbi, T. A. (1989). Program understanding: Challenge for the 1990s. *IBM Systems Journal 28*(2), 294-305.

Deissenboeck, F. & Pizka, M. (2006) Concise and consistent naming. *Software Quality Control*, *14*(3), 261–282.

Della, L., & Clark, D. (2000, December 1). *Teaching object-oriented development with emphasis on pattern application.* Paper presented at the Proceedings of the Australasian conference on Computing education, Melbourne, Australia. doi:10.1145/359369.359378

Derk, M. (2011, October 22) *What makes a programming language popular?: An essay from a historical perspective.* Paper presented at the Proceedings of the 10th SIGPLAN symposium on New ideas, new paradigms, and reflections on programming and software, Portland, OR. doi:10.1145/2089131.2089139

Gopstein, D., Iannacone, J., Yan, Y., DeLong, L., Zhuang, Y., Yeh, M., & Cappos, J. (2017, September 4). *Understanding misunderstandings in source code*. Paper presented at the Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, Paderborn, Germany. doi:10.1145/3106237.3106264

Guerrouj, L. (2013, May 18). *Normalizing source code vocabulary to support program comprehension and software quality*. Paper presented at the Proceedings of the 2013 International Conference on Software Engineering, San Francisco, CA. doi:10.1109/ICSE.2013.6606723

Kernighan, B. W., & Ritchie, D. M. (1988). *The C programming language* (2nd ed.). Englewood Cliffs, NJ: Prentice-Hall.

Kuhn, A., Ducasse, S., & Girba, T. (2007). Semantic clustering: Identifying topics in source code. *Information and Software Technology 49,* 230-243. doi: 10.1016

Li, X., & Prasad, C. (2005, October 20). *Effectively teaching coding standards in programming.* Paper presented at the Proceedings of the 6th Conference on Information Technology Education, Newark, NJ.doi:10.1145/1095714.1095770

Lientz, B. P., Swanson, E. B., & Tompkins, G. E. (1978, June). Characteristics of application software maintenance. *Communications of the ACM 21* (6), 466-471.

Schober, D., Smith, B., Lewis, S. E., Kusnierczyk, W., Lomax, J., Mungall, C., . . . Sansone, S. (2009, April 27). Survey-based naming conventions for use in OBO Foundry ontology development. *BMC Bioinformatics*, *10*(1), 125. doi:10.1186/1471-2105-10-125

Sharif, B., &Maletic, J. I. (2010, June 30). *An eye tracking study on camelcase and under_score identifier styles*. Paper presented at the ICPC '10 Proceedings of the 2010 IEEE 18th International Conference on Program Comprehension, Washington, DC. 196-205. doi:10.1.1.381.2034

Shipley, C. LTC, & Burks, R. CPT (1997) *Security operations training in the brigade support area (BSA)*. NTC GOLDMINERS' Tactics, Techniques and Procedures for Combat Service Support, 97-14. Retrieved March 28, 2018, from https://www.globalsecurity.org/military/library/report/call/call_97-14_ntcart1.htm

APPENDIX A

TEXAS A&M UNIVERSITY CENTRAL TEXAS

Thesis Research: Programming Standards

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and the readability and comprehension of naming conventions.

Date: Monday, Dec 11 to Thursday, Dec 14, 2017
Time: Starting every hour from 10 AM to 2 PM
Location: TAMUCT Computer Lab, WH 104

Participation will take 10 – 15 minutes to complete.

You must be at least ***18 years*** of age or older and have completed at least ***1 Computer Science course*** in your college based education.

Please Note: This research is completely voluntary, you may leave at any point during the experiment. You will also receive no incentives for your participation.

APPENDIX B

CONSENT FORM

Naming Conventions In Object Oriented Programming

Introduction

The purpose of this form is to provide you information that may affect your decision as to whether or not to participate in this research study. If you decide to participate in this study, this form will also be used to record your consent. Consent is confirmed by your participation on the survey and the protocol.

You have been asked to participate in a research project studying the need and usefulness of a naming convention standard in object-oriented programming. The purpose of this study is to determine if there is a need for a naming convention standard and if a more detailed naming convention is more effective in comprehending the code. You were selected to be a possible participant because of your experience in programming and enrollment at Texas A&M University - Central Texas.

What will I be asked to do?

If you agree to participate in this study, you will be asked to review two pieces of code at a time, one with a naming convention and one without. After reading one, you will then be asked to answer a series of questions related to the comprehension of the code. Once you have completed one, you may only then proceed to the next questionnaire. After both have been completed, a short five question reflection will be given and asked to be completed. This study will take approximately 10 to 30 minutes.

What are the risks involved in this study?

The risks associated in this study are minimal, and are not greater than risks ordinarily encountered in daily life.

What are the possible benefits of this study?

You will receive no direct benefit from participating in this study; however, the benefits of this research is that organizations that focus on programming will know if a detailed naming convention is more effective. Also based on the research findings, organizations and academicians will know whether to start implementing and teaching a naming convention standard.

Do I have to participate?

No. Your participation is voluntary. You may decide not to participate or to withdraw at any time without your current or future relations with Texas A&M University-Central Texas being affected.

Who will know about my participation in this research study?

This study is confidential. No identifiers linking you to this study will be included in any sort of report that might be published. Research records will be stored securely and only Stephen Tilzey and Dr. Anitha Chennamaneni, will have access to the records.

Whom do I contact with questions about the research?

If you have questions regarding this study, you may contact: Stephen Tilzey, 254-319-4337, or at StephenKicker@aol.com. You may also contact Dr. Chennamaneni at anitha.chennamaneni@tamuct.edu.

Whom do I contact about my rights as a research participant?
This research study has been reviewed by the Research Compliance Officer and/or the Institutional Review Board at Texas A&M University-Central Texas. For research-related problems or questions regarding your rights as a research participant, you can contact Walter Murphy, Research Compliance Officer, at (254) 519-5761 or murphyw@tamuct.edu.

Agreement to Participate:
Please be sure you have read the above information, asked questions and received answers to your satisfaction. You must be **over the age of 18** and have taken at least **1 programming course** in order to participate in this research. Please do not complete the survey if you do not meet these requirements. Thank you and have a wonderful day.

APPENDIX C

TEXAS A&M UNIVERSITY CENTRAL TEXAS

Time:_____

FORM C1a

Programming Standards Questionnaire:

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and comprehension of naming conventions. Information is requested only for comparative study analysis and the reporting on questionnaires will be anonymous.

Please read the following lines of code. For each snippet, answer each column on a scale of 1 to 5 with either 1 "least" or 5 "most" for how you would rate the code based on the given factor.

	Readability	Simplicity	Understanding
string str_firstName; str_firstName = "John"; Console.Out.WriteLine(str_firstName + " Doe");			

If (int_value.Contains(int_valueSearch)){
Console.Out.WriteLine("Yes");
}			

for(int int_loopIndex = 0; int_loopIndex <10; int_loopIndex++) {
Console.Out.WriteLine("value: " + int_loopIndex) ;
}			

Public Void Pub_CalculatePay() {
double dec_salary = dec_hourlyPay * int_hoursWorked;
}			

If (str_name1.Equals(str_name2)){
Console.Out.WriteLine(str_true);
}
else {
Console.Out.WriteLine(str_false);
}			

string str_substring = str_original.Substring(2);			

```
For (int int_loopIndex = int_zero; int_loopIndex < (int_maximum-1); int_loopIndex ++) {
   ary_int_decrementing [int int_loopIndex ] = int_startingValue – int int_loopIndex ;
   ary_int_incrementing [int int_loopIndex ] = int_startingValue + int int_loopIndex ;
}
```

In the code snippet above, briefly describe what the code is doing:

Do you find the naming convention helps in reading and understanding the code snippet above? Yes ☐ No ☐

```
Public Void Pub_CalculatePay() {
   double dec_salary = dec_hourlyPay * int_hoursWorked;
}
```

In the code snippet above, briefly describe the purpose of the variable "int_hoursWorked":
If not enough information is available to determine variable purpose, please state "Not Determinable"

Do you find the naming convention helps in determining the variable's purpose in the code above? Yes ☐ No ☐

TEXAS A&M UNIVERSITY CENTRAL TEXAS

Time:_____

FORM C1b

Programming Standards Questionnaire:

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and comprehension of naming conventions. Information is requested only for comparative study analysis and the reporting on questionnaires will be anonymous.

Please read the following lines of code. For each snippet, answer each column on a scale of 1 to 5 with either 1 "least" or 5 "most" for how you would rate the code based on the given factor.

	Readability	Simplicity	Understanding
string x; x = "John"; Console.Out.WriteLine(x + " Doe");			

If (num.Contains(var1)){
Console.Out.WriteLine("Yes");
}			

for(int a = 0; a <10; a++) {
Console.Out.WriteLine("value: " + a) ;
}			

Public Void Calc() {
double num = money * var1;
}			

If (n3.Equals(n2)){
Console.Out.WriteLine(z);
}
else {
Console.Out.WriteLine(y);
}			

string sbstr = st.Substring(2);			

```
For (int c = ZERO; c < (z-1); c++) {
  numberarray[c] = var1 – c;
  arrayofnumbers[c] = var1 + c;
}
```

In the code snippet above, briefly describe what the code is doing:

Do you find the lack of a naming convention helps in reading and understanding the code snippet above? Yes ☐ No ☐

```
Public Void Calc() {
  double num = money * var1;
}
```

In the code snippet above, briefly describe the purpose of the variable "var1":
If not enough information is available to determine variable purpose, please state "Not Determinable"

Do you find the lack of a naming convention helps in determining the variable's purpose in the code? Yes ☐ No ☐

TEXAS A&M UNIVERSITY CENTRAL TEXAS

Time:_____

FORM C2a

Programming Standards Questionnaire:

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and comprehension of naming conventions. Information is requested only for comparative study analysis and the reporting on questionnaires will be anonymous.

Please read the following lines of code. For each snippet, answer each column on a scale of 1 to 5 with either 1 "least" or 5 "most" for how you would rate the code based on the given factor.

	Readability	Simplicity	Understanding

int int_numOfHours = 1;
int_numOfHours = 4 + int_numOfHours;
Console.Out.WriteLine("Number of hours: " + int_numOfHours);			

If (int_cookiesBaked < int_cookiesNeeded){
Console.Out.WriteLine("Not enough.");
}			

do { Int_loopIndex +=1; }while(Int_loopIndex<10);			

Public int Pub_CalculateArea() {
int int_area = int_length * int_width; return int_area;
}			

If (int_testScore >= 70){
bln_passed = true;
}
else {
bln_failed = true;
}			

string str_fullName = str_firstName + " " + str_lastName; Console.Out.WriteLine(str_fullName);			

```
Public int Pub_CalculatePerimeter() {
   int int_perimeter = int_width + int_width + int_length + int_length;
   return int_perimeter;
}
```

In the code snippet above, briefly describe what the code is doing:
If not enough information is available to determine, please state "Not Determinable"

Do you find the naming convention helps in reading and understanding the code snippet above? Yes ☐ No ☐

```
Int[] ary_int_quizScores = new int[30];
```

In the code snippet above, list a few values that would most likely be found in the "ary_int_quizScores" array:
If not enough information is available to determine variable purpose, please state "Not Determinable"

Do you find the naming convention helps in determining possible values in the array? Yes ☐ No ☐

TEXAS A&M UNIVERSITY CENTRAL TEXAS

Time:_____

FORM C2b

Programming Standards Questionnaire:

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and comprehension of naming conventions. Information is requested only for comparative study analysis and the reporting on questionnaires will be anonymous.

Please read the following lines of code. For each snippet, answer each column with either 1 "least" or 5 "most" for how you would rate the code based on the given factor.

	Readability	Simplicity	Understanding

int hr = 1;
hr = 4 + hr;
Console.Out.WriteLine("Number of hours: " + hr);			

If (cks < need){
Console.Out.WriteLine("Not enough.");
}			

do {
var30 +=1;
}while(var30 <10);			

Public int Area() {
int a = x * y; return a;
}			

If (scr >= 70){
ps = true;
}
else {
fl = true;
}			

string n = fn + " " + ln; Console.Out.WriteLine(n);			

```
Public int Calc() {
   int p = x + x + y + y;
   return p;
}
```

In the code snippet above, briefly describe what the code is doing:
If not enough information is available to determine, please state "Not Determinable"

Do you find the lack of a naming convention helps in reading and understanding the code snippet above? Yes ☐ No ☐

```
Int[] qzscrs = new int[30];
```

In the code snippet above, list a few values that would most likely be found in the "qzscrs" array:
If not enough information is available to determine variable purpose, please state "Not Determinable"

Do you find the lack of a naming convention helps in determining possible values in the array? Yes ☐ No ☐

APPENDIX D

TEXAS A&M UNIVERSITY CENTRAL TEXAS

Time:_____

FORM J1a

Programming Standards Questionnaire:

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and comprehension of naming conventions. Information is requested only for comparative study analysis and the reporting on questionnaires will be anonymous.

Please read the following lines of code. For each snippet, answer each column on a scale of 1 to 5 with either 1 "least" or 5 "most" for how you would rate the code based on the given factor.

	Readability	Simplicity	Understanding
String str_firstName; str_firstName = "John"; System.out.println(str_firstName + " Doe");			

If (int_value.contains(int_valueSearch)){
System.out.println("Yes");
}			

for(int int_loopIndex = 0; int_loopIndex <10; int_loopIndex++) {
System.out.println("value: " + int_loopIndex) ;
}			

Public Void Pub_CalculatePay() {
double dec_salary = dec_hourlyPay * int_hoursWorked;
}			

If (str_name1.Equals(str_name2)){
System.out.println(str_true);
}
else {
System.out.println(str_false);
}			

String str_substring = str_original.substring(2);			

```
For (int int_loopIndex = int_zero; int_loopIndex < (int_maximum-1); int_loopIndex ++) {
   ary_int_decrementing [int int_loopIndex ] = int_startingValue – int int_loopIndex ;
   ary_int_incrementing [int int_loopIndex ] = int_startingValue + int int_loopIndex ;
}
```

In the code snippet above, briefly describe what the code is doing:

Do you find the naming convention helps in reading and understanding the code snippet above? Yes ☐ No ☐

```
Public Void Pub_CalculatePay() {
   double dec_salary = dec_hourlyPay * int_hoursWorked;
}
```

In the code snippet above, briefly describe the purpose of the variable "int_hoursWorked":
If not enough information is available to determine variable purpose, please state "Not Determinable"

Do you find the naming convention helps in determining the variable's purpose in the code above? Yes ☐ No ☐

TEXAS A&M UNIVERSITY CENTRAL TEXAS

Time:_____

FORM J1b

Programming Standards Questionnaire:

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and comprehension of naming conventions. Information is requested only for comparative study analysis and the reporting on questionnaires will be anonymous.

Please read the following lines of code. For each snippet, answer each column on a scale of 1 to 5 with either 1 "least" or 5 "most" for how you would rate the code based on the given factor.

	Readability	Simplicity	Understanding
String x; x = "John"; System.out.println(x + " Doe");			

If (num.contains(var1)){
System.out.println("Yes");
}			

for(int a = 0; a <10; a++) {
System.out.println("value: " + a) ;
}			

Public Void Calc() {
double num = money * var1;
}			

If (n3.equals(n2)){
System.out.println(z);
}
else {
System.out.println(y);
}			

String sbstr = st.substring(2);			

```
For (int c = ZERO; c < (z-1); c++) {
  numberarray[c] = var1 – c;
  arrayofnumbers[c] = var1 + c;
}
```

In the code snippet above, briefly describe what the code is doing:

Do you find the lack of a naming convention helps in reading and understanding the code snippet above? Yes ☐ No ☐

```
Public Void Calc() {
  double num = money * var1;
}
```

In the code snippet above, briefly describe the purpose of the variable "var1":
If not enough information is available to determine variable purpose, please state "Not Determinable"

Do you find the lack of a naming convention helps in determining the variable's purpose in the code? Yes ☐ No ☐

TEXAS A&M UNIVERSITY CENTRAL TEXAS

Time:_____

FORM J2a

Programming Standards Questionnaire:

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and comprehension of naming conventions. Information is requested only for comparative study analysis and the reporting on questionnaires will be anonymous.

Please read the following lines of code. For each snippet, answer each column on a scale of 1 to 5 with either 1 "least" or 5 "most" for how you would rate the code based on the given factor.

	Readability	Simplicity	Understanding

int int_numOfHours = 1;
int_numOfHours = 4 + int_numOfHours;
System.out.println("Number of hours: " + int_numOfHours);			

If (int_cookiesBaked < int_cookiesNeeded){
System.out.println("Not enough.");
}			

do { Int_loopIndex +=1; }while(Int_loopIndex<10);			

Public int Pub_CalculateArea() {
int int_area = int_length * int_width; return int_area;
}			

If (int_testScore >= 70){
bln_passed = true;
}
else {
bln_failed = true;
}			

String str_fullName = str_firstName + " " + str_lastName; System.out.println(str_fullName);			

```
Public int Pub_CalculatePerimeter() {
   int int_perimeter = int_width + int_width + int_length + int_length;
   return int_perimeter;
}
```

In the code snippet above, briefly describe what the code is doing:
If not enough information is available to determine, please state "Not Determinable"

Do you find the naming convention helps in reading and understanding the code snippet above? Yes ☐ No ☐

```
Int[] ary_int_quizScores = new int[30];
```

In the code snippet above, list a few values that would most likely be found in the "ary_int_quizScores" array:
If not enough information is available to determine variable purpose, please state "Not Determinable"

Do you find the naming convention helps in determining possible values in the array? Yes ☐ No ☐

TEXAS A&M UNIVERSITY CENTRAL TEXAS

Time:_____

FORM J2b

Programming Standards Questionnaire:

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and comprehension of naming conventions. Information is requested only for comparative study analysis and the reporting on questionnaires will be anonymous.

Please read the following lines of code. For each snippet, answer each column with either 1 "least" or 5 "most" for how you would rate the code based on the given factor.

	Readability	Simplicity	Understanding

int hr = 1;
hr = 4 + hr;
System.out.println("Number of hours: " + hr);			

If (cks < need){
System.out.println("Not enough.");
}			

do {
var30 +=1;
}while(var30 <10);			

Public int Area() {
int a = x * y; return a;
}			

If (scr >= 70){
ps = true;
}
else {
fl = true;
}			

String n = fn + " " + ln; System.out.println(n);			

```
Public int Calc() {
   int p = x + x + y + y;
   return p;
}
```

In the code snippet above, briefly describe what the code is doing:
If not enough information is available to determine, please state "Not Determinable"

Do you find the lack of a naming convention helps in reading and understanding the code snippet above? Yes ☐ No ☐

```
Int[] qzscrs = new int[30];
```

In the code snippet above, list a few values that would most likely be found in the "qzscrs" array:
If not enough information is available to determine variable purpose, please state "Not Determinable"

Do you find the lack of a naming convention helps in determining possible values in the array? Yes ☐ No ☐

APPENDIX E

TEXAS A&M UNIVERSITY CENTRAL TEXAS

Time:_____

FORM V1a

Programming Standards Questionnaire:

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and comprehension of naming conventions. Information is requested only for comparative study analysis and the reporting on questionnaires will be anonymous.

Please read the following lines of code. For each snippet, answer each column on a scale of 1 to 5 with either 1 "least" or 5 "most" for how you would rate the code based on the given factor.

	Readability	Simplicity	Understanding
Dim str_firstName As String str_firstName = "John" Console. Writeline(str_firstName & " Doe")			

If (int_value.Contains(int_valueSearch)) Then
Console.WriteLine("Yes")
End If			

For int_loopIndex As Integer = 0 To 10 Console.WriteLine("value: " & int_loopIndex) Next			
Sub Sub_CalculatePay() Dim dec_salary As Decimal = dec_hourlyPay * int_hoursWorked End Sub			

If (str_name1.Equals(str_name2)) Then
Console.WriteLine(str_true)
Else
Console.WriteLine(str_false)
End If			

Dim str_substring As String = str_original.Substring(2)			

```
For int_loopIndex As Integer = int_zero To (int_maximum - 1)
   ary_int_decrementing(int_loopIndex) = int_startingValue - int_loopIndex
   ary_int_incrementing(int_loopIndex) = int_startingValue+ int_loopIndex
Next
```

In the code snippet above, briefly describe what the code is doing:

Do you find the naming convention helps in reading and understanding the code snippet above? Yes ☐ No ☐

```
Sub Sub_CalculatePay()
   Dim dec_salary As Decimal = dec_hourlyPay * int_hoursWorked
End Sub
```

In the code snippet above, briefly describe the purpose of the variable "int_hoursWorked":

If not enough information is available to determine variable purpose, please state "Not Determinable"

Do you find the naming convention helps in determining the variable's purpose in the code above? Yes ☐ No ☐

TEXAS A&M UNIVERSITY CENTRAL TEXAS

Time:_____

FORM V1b

Programming Standards Questionnaire:

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and comprehension of naming conventions. Information is requested only for comparative study analysis and the reporting on questionnaires will be anonymous.

Please read the following lines of code. For each snippet, answer each column on a scale of 1 to 5 with either 1 "least" or 5 "most" for how you would rate the code based on the given factor.

	Readability	Simplicity	Understanding
Dim x As String x = "John" Console. Writeline(x & " Doe")			

If (num.Contains(var1)) Then
Console.WriteLine("Yes")
End If			

For a As Integer = 0 To 10 Console.WriteLine("value: " & a) Next			
Sub Calc() Dim num As Decimal = money * var1 End Sub			

If (n3.Equals(n2)) Then
Console.WriteLine(z)
Else
Console.WriteLine(y)
End If			

Dim sbstr As String = st.Substring(2)			

```
For c As Integer = ZERO To (z - 1)
  numberarray(c) = var1 - c
  arrayofnumbers(c) = var1 + c
Next
```

In the code snippet above, briefly describe what the code is doing:

Do you find the lack of a naming convention helps in reading and understanding the code snippet above? Yes ☐ No ☐

```
Sub Calc()
  Dim num As Decimal = money * var1
End Sub
```

In the code snippet above, briefly describe the purpose of the variable "var1":

If not enough information is available to determine variable purpose, please state "Not Determinable"

Do you find the lack of a naming convention helps in determining the variable's purpose in the code? Yes ☐ No ☐

TEXAS A&M UNIVERSITY CENTRAL TEXAS

Time:_____

FORM V2a

Programming Standards Questionnaire:

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and comprehension of naming conventions. Information is requested only for comparative study analysis and the reporting on questionnaires will be anonymous.

Please read the following lines of code. For each snippet, answer each column on a scale of 1 to 5 with either 1 "least" or 5 "most" for how you would rate the code based on the given factor.

	Readability	Simplicity	Understanding
Dim int_numOfHours As Integer = 1 int_numOfHours = 4 + int_numOfHours Console.WriteLine("Number of hours: " & int_numOfHours)			

If (int_cookiesBaked < int_cookiesNeeded) Then
  Console.WriteLine("Not enough.")
End If			

Do Int_loopIndex +=1; Loop Until Int_loopIndex>10			

Function Fun_CalculateArea() as Integer
  Dim int_area As Integer= int_length * int_width
  return int_area
End Function			

If (int_testScore >= 70)Then
  bln_passed = True
Else
  bln_failed = True
End If			

Dim str_fullName As String = str_firstName & " " & str_lastName Console.WriteLine(str_fullName)			

```
Function Fun_CalculatePerimeter() As Integer
  Dim int_perimeter As Integer = int_width + int_width + int_length + int_length
  return int_perimeter
End Function
```

In the code snippet above, briefly describe what the code is doing:
If not enough information is available to determine, please state "Not Determinable"

Do you find the naming convention helps in reading and understanding the code snippet above? Yes ☐ No ☐

```
Dim ary_int_quizScores(30) As Integer
```

In the code snippet above, list a few values that would most likely be found in the "ary_int_quizScores" array:
If not enough information is available to determine variable purpose, please state "Not Determinable"

Do you find the naming convention helps in determining possible values in the array? Yes ☐ No ☐

TEXAS A&M UNIVERSITY CENTRAL TEXAS

Time:_____

FORM V2b

Programming Standards Questionnaire:

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and comprehension of naming conventions. Information is requested only for comparative study analysis and the reporting on questionnaires will be anonymous.

Please read the following lines of code. For each snippet, answer each column with either 1 "least" or 5 "most" for how you would rate the code based on the given factor.

	Readability	Simplicity	Understanding
Dim hr As Integer = 1 hr = 4 + hr Console.WriteLine("Number of hours: " & hr)			

If (cks < need) Then
  Console.WriteLine("Not enough.")
End If			

Do var30 +=1; Loop Until var30>10			

Function Area() as Integer
  Dim a As Integer= x * y
  return a
End Function			

If (scr >= 70)Then
  ps = True
Else
  fl = True
End If			

Dim n As String = fn & " " & ln Console.WriteLine(n)			

```
Function Calc () As Integer
  Dim p As Integer = x + x + y + y
  return p
End Function
```

In the code snippet above, briefly describe what the code is doing:
If not enough information is available to determine, please state "Not Determinable"

Do you find the lack of a naming convention helps in reading and understanding the code snippet above? Yes ☐ No ☐

```
Dim qzscrs(30) As Integer
```

In the code snippet above, list a few values that would most likely be found in the "qzscrs" array:
If not enough information is available to determine variable purpose, please state "Not Determinable"

Do you find the lack of a naming convention helps in determining possible values in the array? Yes ☐ No ☐

APPENDIX F

TEXAS A&M UNIVERSITY CENTRAL TEXAS

FORM R1

Programming Standards Questionnaire: Reflection & Demographics

My name is Stephen Tilzey, a current graduate student of Texas A&M University – Central Texas. I am working on a research thesis to study the standards in object oriented programming and comprehension of naming conventions.

Information is requested only for comparative study analysis and the reporting on questionnaires will be anonymous.

Please read the questions and responses carefully, and select the value of 1-7 that most accurately reflects your answer; *1 being strongly agree – 7 being strongly disagree.*

Reflection Questions:	1-7
There should be a naming convention standard in object-oriented programming.	
It was easier to read the code that did not follow a naming convention.	
It was easier to read the code that followed a naming convention.	

Demographic Questions: ***Please the select the most appropriate response.***

1a. Gender ***(Select One)***
[ ] Male [ ] Female

1b. Age: ***(Select One)***
[ ] < 18 [ ] 18-24 [ ] 25-30 [ ] 31-45 [ ] 45>

1c. Current Occupation: ***(Select The Most Applicable Response)***
[ ] Student [ ] Employed [ ] Un-employed [ ] Retired

1d. Current Education:
[ ] High School [ ] Associates [ ] Bachelors [ ] Masters [ ] PhD
Please list your major(s):

1e. Programming Experience: ***(Select One)***
[ ] <1 yr [ ] 2 – 3 yrs [ ] 4 – 5 yrs [ ] >6 yrs

1f. How many programming languages do you know?:
[ ] 0 [ ] 1 [ ] 2 [ ] 3 [ ] >4
Please list them in the space below:

Thank you for participating in this research. Your responses will be crucial in understanding the need and usefulness of naming convention standards. We hope you have a wonderful day.