Understanding Humans' Cognitive Processes During Computational Thinking Through Cognitive Science

Kiran Datwani, Michael-Brian C. Ogawa(✉), and Martha E. Crosby(✉)

Department of Information and Computer Sciences,
University of Hawaii at Manoa, Hawaii, USA
{kiran25,ogawam,crosby}@hawaii.edu

Abstract. Human-computer interaction is a diverse field covering disciplines such as computer technology, human factors, and cognitive science to name a few. Over the past several years, the information age has developed to incorporate a society that intentionally and unintentionally interacts with computing technology every day. The field of computational thinking in human-computer interaction is expanding and incorporating multidisciplinary fields such as psychology and software principles. Research has been conducted in the past regarding the background, social impact and innovation, and a new direction in social computing/ issues in HCI. HCI is a diverse, expansive field covering many aspects and disciplines in computer science, the humanities, and others. Computational thinking, a subfield of HCI, explores the way humans process problems, and use problem-solving skills and analogies to solve complex, or seemingly difficult problems (Wing 2006). This research project will be conducted to understand computational thinking in people, along with determining the existing relationship between cognitive science and HCI.

Keywords: Computational thinking · Cognitive science · Human-computer interaction

1 Introduction

1.1 Research Questions

What facets/aspects of cognitive science help researchers in human-computer interaction (HCI) understand how people interact with computers? Are there existing relationships between cognitive science at an individual level in problem-solving and HCI at a group or community level?

1.2 Specific Aims

The questions will be answered by the following specific aims. First, to understand different peoples' computational thinking, and the amount of time it takes for a person

D. D. Schmorrow and C. M. Fidopiastis (Eds.): HCII 2022, LNAI 13310, pp. 242–260, 2022.
https://doi.org/10.1007/978-3-031-05457-0_20

depending on the level of their computer literacy (e.g., programming novices, computer science majors, public) and second to identify, understand, and solve a problem. Secondly, to identify the level of expertise of participants, and determining different approaches to be taken for participants to understand the bigger picture of a program and solve problems with a stronger grasp of computing knowledge and literacy.

1.3 Significance of Study

Computing technology has revolved rapidly in the last several decades, however, the relationship between computing technology and the cognitive science when processing information on computing devices is vague. Computational thinking is a portion of computer science that is used by people on a daily basis to take relevant smaller steps to understand and solve larger problems. That being said, it is rare for people to correlate computational thinking with people's cognitive processes, and is seen more as a way computers are programmed, and the steps computer scientists take to write programs (Wing 2006). In this research study, I will be using cognitive science methodology to better understand the cognitive processes that facilitate seeking, filtering, and shaping relevant information to perform tasks that involve computational thinking. With a stronger understanding of computational thinking in individuals of various backgrounds, it can be determined how the problem-solving process takes place depending on an individual's background. Alongside that, I will be developing a stronger understanding of computational thinking and its relationship with HCI and cognitive science.

2 Introduction Background Information and Literature Review

Human-computer interaction is a comprehensive field covering disciplines such as computer technology, human factors, and cognitive science to new a few. Over the past several years, the information age has developed to incorporate a society that intentionally and unintentionally interacts with computing technology everyday. The field is expanding and incorporating multidisciplinary fields such as psychology and software principles. The literature review was conducted to further develop knowledge of human-computer interaction, cognitive science, and problem-solving for Senior Honors Project preparation. The terms I used to retrieve the following documents are cognitive science, psychology, human-computer interaction, society AND the information age, and the search engines used were Google Scholar and PubMed. My topics were divided into 5 parts: (1) Background of HCI, (2) Background on cognitive science, (3) Social impact, innovation, and computational thinking, (4) New direction in social computing/issues in HCI, and (5) Implications and impacts of cognitive science in Computer Science. Research in this topic is contributed by strong interest in the connection between humans and computers, and the aspiration to develop a project that will allow in contribution to this topic. As of now, there are several articles available written solely about human-computer interaction, a few written about the cognitive science factors of HCI during the information era, and this paper will reflect the background needed for this study.

2.1 Background on HCI

Human-computer interaction, which is also known as HCI, is a diverse, integrative field that encompasses aspects of human factors, system design, technical writing, and programming abilities. Cognitive science, on the other hand, while also an inclusive field overlapping in disciplines such as psychology, linguistics and philosophy, have little means in connecting the disciplines as a common idea. HCI is said to be a role that plays in connecting the disciplines of cognitive science to allow it to become more cohesive; HCI may be able to contribute to a better understanding of cognitive science and its diverse fields. Sharing a similar history, human-computer interaction and cognitive science were formed and developed during the time of World War II, to train soldiers to fight in the war in faster, better methods. It was late after World War II that research began being conducted on the interactions of humans and computers, which eventually led to HCI becoming a collaboration between psychologists and computer scientists. The aspects of cognitive science are included within the realm of HCI, but what is lacking is their inclusion with each other, and their overall role in HCI research (Boring 2002).

There is high importance in user-friendly systems and the development of usable systems using user interface media. The field of human-computer interaction and the components developed from it are unlike any other field, including the disciplines in engineering and design, because it is a combination of the two. Throughout time, user interfaces have become easier to learn and use by consumers, but far more difficult to program and build by programmers and engineers. The concept of user interfaces brings attention to the understanding that the majority of consumers of technological products and achievements have scarce knowledge of the difficulty of programming. It is important to note the development of design through the conceptual, semantic, syntactic, and lexical levels, and how they are impacted by the user and their ability in under-standing and using the software interface (Butler et al. 1998).

Human-computer interaction has become the visible part of computer science and has in recent years become crucial in creating solutions for difficult problems, such as in areas on direct manipulation interfaces, user interface management systems, task-oriented help and instruction, and computer-supported collaborative work. HCI is defined as a science of design, and is established in order to support and understand humans interacting with technology. HCI in software psychology has origins from several decades ago, and during that time, problems were posed. The two problems that were posed were firstly, describing and supporting design and development work, and secondly, better specifying the role social and behavioral science, which are parts of psychology, should play in HCI (Carrol 1997).

2.2 Background on Cognitive Science

Previously, psychology was dominated by the behaviorist approach. It was much later on that the human information-processing approach known as the cognitive revolution was discovered by the progress in communication system engineering and statistics. This approach focused on gaining knowledge of the human mind, and gave explanations to a human's perceptions and actions. Psychology and society have both been impacted by the information age largely, and science and technology have only continued to grow and expand throughout the years. It has been stated that eventually in time, the era of the information age will cease to exist, with the explanation that no era lasts forever, and thus, will be replaced with a new age in time (Xiong and Proctor 2018).

Individuals are responsible for creating their own unique experiences through interpretations, emotions and their judgments as elaborated by McCarthy and Wright (2005). Experience is thus created by the relationship between the self and the object, and the concept of life as lived and felt is stated by the writers to be personal, constructive, and transformative, and the connection between interior and exterior aspects of experience. In relation to HCI, the aspect that is already integrated within understanding HCI is the approaches of cognitive information processing addressing the efficiency and effectiveness of performance, but the aspect of understanding a person's experience with technology and their relationship, in positive and negative factors, is usually missed. While both are crucial, they are approached in different methodologies; while cognitive information processing is impersonal, and accessed through observation and experimentation, the felt-like experiences are wholly and completely personal (McCarthy and Wright 2005).

2.3 Social Impact, Innovation, and Computational Thinking

Throughout learning, growth, and education, students are strongly encouraged to develop a computational thinking approach, which is the concept of thinking in logical procedural processes. Over the years, a constant thread of the lack, or reduction, of innovation is developed because of the way young people are built to think and complete tasks. By social theories, computation is a major contributor in the lack of innovation. Although it is said that computation is required in order to achieve constant innovation at a steady pace, in the current time and age, computation has actually become the reason that innovation has lessened and impeded (Oren 2011). Although there isn't exact correlation, the reason behind this can be contributed by humans developing lesser skills in the innovation process and step-by-step problem-solving because computing technology has allowed processes to become simpler. A crucial aspect of this study will be to formulate whether stronger skills in computer literacy has benefited rather than harmed their problem-solving skills.

Computational thinking, which are problem-solving skills and thought processes to solve complex problems and write programs, are not only used by software engineers and computer scientists, but by everyone. The concept refers to steps taken in order to analyze, comprehend, and understand how to solve a problem. Computational thinking, along with involving solving problems, includes designing systems, and understanding human behavior, which elaborates on the concepts fundamental to computer science. The field of computer science is encompassed by computational thinking, which includes a range of mental tools. The concept is used in everyday activities, such as preparing your bag for what you need for school/work, losing an item and retracing your steps to find them, or picking a line in the supermarket that will allow you to wait the least amount of time. It takes the idea of a large difficult problem and breaks it down into steps and/or analogies, small pieces of the entire problem, to make it easier to understand and solve the larger picture of the problem. The steps and/or analogies serve as the representation or model of the challenging, relevant aspect of a problem, usually by reduction, embedding, transformation, or simulation. Computational thinking expresses that despite not having every detail or not understanding a problem wholly, or the end goal completely, it is possible to take a large problem, and still be able to use, and modify it to provide a result or solution. The concept of computational thinking is effectively implemented in everyone's lives, not only computer scientists, and it represents the way humans think, and comprehend and solve problems, rather than how a computer is programmed to solve tasks (Wing 2006) (Fig. 1).

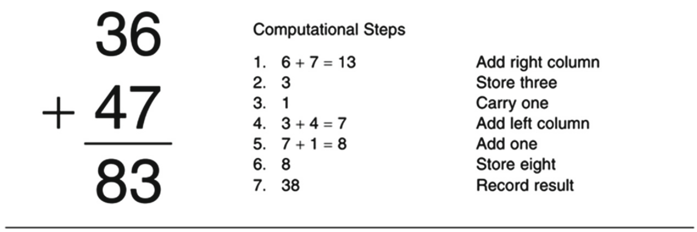

Fig. 1. The computational steps represent a simple arithmetic problem that is being solved through computational thinking (Friedenberg and Silverman 2006).

2.4 New Direction in Social Computing/issues in HCI

Despite plentiful research and knowledge accumulated in HCI, many researchers are continuing to be unaware of issues pertaining to HCI, such as systems being user-friendly and understanding the intentions and motives of the user. Usability issues are experienced both subjectively and collectively by different groups of people in mass populations. A programmer or computer engineer must understand the users' motives and perspectives, emotional and social drives to develop concepts and methodology that would be applicable to locations such as work practices, communities and organizational social structures. One of the most crucial aspects is understanding the design, development and implementation of systems individually and as a subunit of the whole product. The article by Cairns et al. directly points to the direction that collecting qualitative research in HCI is difficult, because it is less on the concept of measuring and producing numbers, and more on understanding the underlying aspects of a certain piece of software or technology, the ways it is used by numerous people, and their thoughts, expressions and emotions of it (Cairns et al. 2008).

A specialist in HCI should have a solid understanding of psychological factors so that they can be able to have the skill or knowledge to relate their observations to observations in other differing fields; it is difficult to be processed by an individual in domains other than psychology. HCI is described as a descriptive, engineering science, as the HCI specialist would be able to provide a level of analysis that would contribute in impacting the design factors on user behavior, as well as provide specific technical details for the overall product (Carroll et al. 1991).

The usage of technological artifacts and applications such as Artificial Intelligence (AI) would become frequent in home, work, and public environments. The interactions would be not only conscious and intentional, but also unintentional. People would be surrounded by and living among visible and invisible technological artifacts, such as numerous examples of AI and the system of transferring information. The eventual goal of the new technological era is continuing to focus on the human and how their needs shape the way technology is manufactured and used. It is first most important to serve the needs of the user. With that mindset, certain problems can arise, such as humans have more requirements and demands than before, and are more attentive and critical of technological artifacts, while at the same time being less optimistic about the outcomes. There are new challenges that are a consequence of human-centered approaches, and how critical issues need to be addressed and solved in order to establish and maintain a trustful and beneficial relationship between humans and technology (Stephanidis et al. 2019) (Fig. 2).

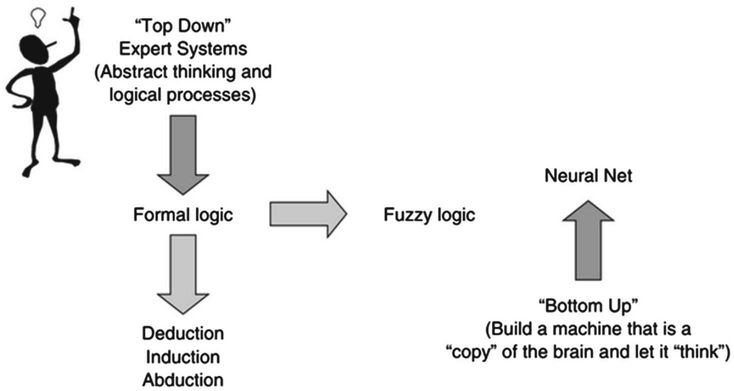

Fig. 2. This diagram given by Friedenberg & Silverman expresses the "top-down" approach of building a machine that "imitates" how humans think, while the "bottom-up" approach is of building a machine and letting it think and make decisions on its' own (Friedenberg and Silverman 2006).

HCI is related to the design, implementation and evaluation of interactive computer-based systems along with the multi-disciplinary study of numerous issues involved in the interaction. The user interface of the computer-based system is the application or system by which the user comes into contact and uses cognitively and physically. Over the years, with the development of the information society, computers have become more than scientific tools, but are becoming devices for productivity enhancement, and integrated into environments in such a way that computers are accessible to all people, regardless of their time and location. The information era has progressed to a place in time where various human activities are mediated by computers and will most likely continue to be so (Stephanidis 2001).

2.5 Implications and Impacts of Cognitive Science in Computer Science

In a study done by BRACElet, it was reported that students in computer science who were able to trace and read code not created by others were able to write similar code in their own style. On the other hand, "students who cannot read a short piece of code and describe it in relational terms are not intellectually well equipped to write similar code" (Lister 2020). This brings insight to the perspective that computer science students who are able to write effective, well-written code are applying problem-solving strategies and fully understanding the issues in hand prior to attempting to solve and program solutions. That being said, it is common for computer science to try and guess solutions without fully tracing, reading, and/or understanding the code presented to them. This can lead to obstacles and issues later on in their professional life where if they are unable to fully understand a problem, the individual will not be able to program a solution for it. As written by Lister, there are three stages to a young programmer learning to understand and write code; the following being Stage 1: Sensorimotor (Pre-Tracing), Stage 2: Pre-operational (Tracing), and Stage 3: Concrete Operational (Post-Tracing) (2020). Throughout the three stages, a novice programmer begins to attempt in reading code to extensively writing their own complete programs.

Recently, Microsoft has developed a low-code, open-source language called the Power Fx, a Power Apps service, that allows an individual to "translate text into code" (Lardinois 2021). Despite the program being fairly easy to use and using natural language as the main form of programming, Microsoft strongly emphases that programmers and novices who wish to use the service "understand the logic of the application they are building" (Lardinois 2021). Despite the replacement of programming knowledge required by the developers, there still needs to be understanding of the concepts and formulas needed to operate and use the service efficiently and effectively. This is similar to the usage of other tools/services such as Excel, PowerBI, and Google Sheets, which also use natural language query functions (2021). What is necessary to acknowledge is that as technology and AI is advancing so are humans' understanding and application of new tools. This is, however, not to disregard that human to halt their problem-solving processes due to technological advancements, but rather, grow their knowledge to easily learn and develop new strategies and solutions while using modern tool sets and technologies.

Eye tracking is one-way researchers use to attempt to understand their participant's cognitive processes and their future actions (Sharafi et al. 2021). Impact on an individual's thought processes including their comprehension, collaboration, emotion, and so forth can be determined using eye trackers. Throughout the relationship between eye trackers and software engineering, it was crucial for the eye trackers to be use correctly to attain the most accurate results. In addition, the data collection must be completed as carefully as possible and analyzed in great detail to fully understand the participants' cognitive processes and intentions. Researchers are able to determine (1) why problems exist in completing a task, (2) the location where participants find key points, (3) whether certain elements or key points prove to be distracting, (4) how participants are guided by designs, layouts, or artifacts whilst completing a task, (5) differences in a participants' efficiency based on their demographics or experience, and (6) whether the participant focused or scanned the details (2021).

In order to understand and solve problems, programmers must have a level of program comprehension established. There are many variations and levels of program comprehension, dependent on individual programmers as well the specific code itself. Programmers in the field who are experts are able to scan code across a computer screen and use key points, features, or semantic cues, also called beacons, to determine the purpose of the program (Siegmund 2017). This approach is known as the top-down approach, specifically to describe cognitive processes based on experience, whilst the bottom-up approach focuses more on finding explanation in individual pieces of the code when semantic cues are missing in the program (2017). In this article, the theme that was explored in great detail can be formed. There has been a deep connection between cognitive science and its relationship to computer science and overall computer literacy. For individuals to use computing technologies well, they need to have a strong grasp of and acquire an understanding of how and why certain programs, languages, and technologies perform the way they do.

3 Methodology and Research Design

3.1 Data Collection (from)

Data was collected in two forms/ways. To fulfill my questions of determining the existing relationship between cognitive science and HCI, and the aspects of cognitive science to interpret computational thinking, I reviewed relevant literature on methodology regarding computational thinking and cognitive science. The literature will also serve as a way to seek insight on development of methodology to form my own for my project. To execute my aims in identifying the level of expertise in computing technology in humans and understanding different people's computational thinking, I developed a series of problems in a software program, in pseudocode, and in natural language, and acquired students' comprehension of said problems to determine their level of computing literacy based on the programs.

3.2 Data Collection (from) Lab Trainings Completed and Ethics Approved

As of the current state, no training was required for this research project. I had received the following certifications from the CITI program: Human Subjects Research (HSR), Conduct of Research, and Information Privacy and Security (IPS) during the Spring of 2021. I had received clearance from the IRB prior to conducting the official research study in November of 2021. The clearance from IRB and the certifications from CITI program were required in order for me to conduct my research. The certifications from the CITI program were discussed with my faculty mentor and committee member during the months of March and April of 2021 and the IRB proposal was discussed in May of 2021. I understand the importance of compliance and ethical standards with the university's standards.

3.3 Resources and Materials

I used articles on computational thinking and methodology of a case study in a cognitive science textbook provided to me by my mentors to develop the methodology of my research project and study. The software I used was Microsoft Excel, which was utilized to develop spreadsheet problems. Lastly, the participants of the project were students of beginner-level ICS courses, such as ICS 101 and 111.

3.4 Role of Researcher

My role as the researcher included a process of three steps. During the first step, I developed the proposal and the execution plan and/or design. This included going back and forth between my mentor and committee member to form the idea and the research process.

Throughout the second step, I collected written work in computational thinking, along with developing a group of problems in the program, Microsoft Excel, pseudocode, and natural language. The problem sets will allow observation in HCI and human usability in computing technology. The goal of the participants was to analyze and comprehend the problem and select the most accurate result in a series of multiple choice questions. During the summer of 2021, a pilot program was conducted for the Teaching Assistants of the ICS 101 class in order to formally develop the problem sets and formulate the final research design for the official study. In the pilot study, there were 9 problems each for spreadsheet, pseudocode, and natural language that was inputted in Laulima as a Quizzes, Tests and Surveys assessment for the 10 Teaching Assistants and two Honors Program student volunteers. The order of the 3 subgroups was in different order based on which test group they were a part of. To elaborate, the first group had questions that were: 3 spreadsheet, 3 pseudocode, and 3 natural language problems, group 2 had 3 pseudocode, 3 natural language, and 3 spreadsheet problems, and lastly, group 3 had 3 natural language, 3 spreadsheet, and 3 pseudocode problems.

Example problems for the question number 1 for group 1, group 2, and group 3 tests developed on Google Forms respectively were the following:

Pseudocode:

```
If student's grade is greater than or equal to 90
     Print "A"
Else if student's grade is greater than or equal to 80
     Print "B"
Else if student's grade is greater than or equal to 70
     Print "C"
Else if student's grade is greater than or equal to 60
     Print "D"
Else
     Print "F"
```

Question: If you type in 67, what would your result be?

Natural Language:

```
If the student's grade is at least 90, print out an A. If it
is below 90, but at least 80, print out a B. If it is below 80,
but at least 70, print out a C. If it is below 70, but at
least 60, print out a D. If not, print a F.
```

Question: If you type in 67, what would your result be?

Microsoft Excel (Spreadsheet):

	A	B	C
1	Name	Score	Grade
2	Student A	86	
3	Student B	77	
4	Student C	52	
5	Student D	100	
6	Student E	91	
7	Student F	82	
8	Student G	67	
9	Student H	73	
10	Student I	85	
11	Student J	97	

```
fx  =IF(B9>=90,"A",IF(B9>=80,"B",IF(B9>=70,"C",IF(B9>=60,"D","F"))))
```

Question: You have the following formula located in cell C8, what result does this function provide, given the information that you have?

Several insights were gained from the pilot study that served in forming the problem sets for the official study. The participants of the project were students of beginner-level ICS courses, such as ICS 101 and 111. There were 101 students for ICS 101 and 11 students for ICS 111 that partook in the study. In the official study, which was created on Google Forms instead of Laulima, there were 9 problems each for spreadsheet, pseudocode, and natural language, and the group of participants were divided into three test groups where they received 3 different problems of each group set. To elaborate, in the ICS 101 section, group 1 received 3 pseudocode, 3 natural language, and 3 spreadsheet problems, group 2 received 3 natural language, 3 spreadsheet, and 3 pseudocode problems, and group 3 received 3 spreadsheet, 3 pseudocode, and 3 natural language problems. The questions remained the same for all 9 problems, however, since there were 3 versions of each question in spreadsheet, pseudocode, and natural language, there was a total of 27 problems created and 9 questions each for the three groups in ICS 101 and three groups in ICS 111. That said, ICS 101 and ICS 111 groups had the same problem sets, such as group 1 for 101 and 111 both had 3 pseudocode problems, then 3 natural language problems, and lastly, 3 Microsoft Excel problems. The questions were written in multiple choice format, with one answer being the correct answer in all.

Finally, during the final step, I collected the data, analyzed the data results, and developed the conclusion.

Example problems for the question number 9 for group 1, group 2, and group 3 tests developed on Google Forms respectively were the following:

Microsoft Excel (Spreadsheet):

	A	B	C	D
1	Calories Chart			
2				
3	DAY	Calories Eaten	Calories Burned	Workout More the Next Day?
4	Sunday	2100	1500	
5	Monday	1500	1200	
6	Tuesday	1200	1200	
7	Wednesday	1800	1900	
8	Thursday	1700	1800	
9	Friday	2200	2000	
10	Saturday	2500	1200	

fx =IF(B6>C6,"Yes","No")

Question: You have the following formula located in cell D6, what result does this function provide, given the information that you have?

Pseudocode:

```
If I eat more calories than I burn
     Print "I will work out more the next day"
Else Print "Do not work out"
```

Question: If I ate 1200 calories and burned 1200 calories today, what would my result be?

Natural Language:

```
If I eat more calories than I burn, I will work out more the
next day. Otherwise, I will not work out more the next day.
```

Question: If I ate 1200 calories and burned 1200 calories today, what would my result be?

4 Data Analysis

The results were analyzed to determine the accuracy rate and the time duration on each program. It was essential to understand the correlation between the correctness of the numbers and the amount of time taken on average by students. With the results, it could determine the program comprehension on average of the students based on the program with high performance, and to identify, based on their knowledge and literacy, their process in understanding and solving problems by computational thinking.

An item analysis was conducted using ANOVA: single factor tests (1) to compare the variance in the three programs within the three groups with their time duration and (2) to compare the variance in the three groups within the three programs with their time duration. The data analysis was conducted on only the official study for ICS 101 since it was a larger set, and more information could be obtained.

4.1 Section 1: Analysis Based on Program

Microsoft Excel

See Tables 1 and 2.

Table 1. Average for correctness of problems for Group 1, 2, and 3 on Microsoft Excel was 0.728395062, 0.888888889, and 0.733333333 respectively, with Group 2 with the highest performance. Group 1 had the third set of problems in Microsoft Excel, while Group 2 had their second set of problems, and Group 3 had their first set of problems in Microsoft Excel.

ANOVA						
Source of variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between groups	0.595803284	2	0.297901642	4.663574614	0.011622715	3.089203013
Within groups	6.260082305	98	0.063878391			
Total	6.855885589	100				

Table 2. Average for the time taken for Group 1, 2, and 3 on Microsoft Excel was 46 s, 45 s, and 1 min and 2 s respectively. Group 2 took the least amount of time on average but still attained the highest amount of correct answers on average.

ANOVA						
Source of variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between groups	8.52024E–07	2	4.26012E–07	2.619700453	0.077535907	3.082014501
Within groups	1.72376E–05	106	1.62619E–07			
Total	1.80896E–05	108				

Pseudocode

See Tables 3 and 4.

Table 3. Average for correctness of problems for Group 1, 2, and 3 on Microsoft Excel was 0.975308642, 0.743589744, and 0.876190476 respectively, with Group 1 with the highest performance. Group 1 had the first set of problems in Pseudocode, while Group 2 had their third set of problems, and Group 3 had their second set of problems in Pseudocode.

ANOVA						
Source of variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between groups	0.889348634	2	0.444674317	10.07033197	0.000105319	3.089203013
Within groups	4.327373038	98	0.044156868			
Total	5.216721672	100				

Table 4. Average for the time taken for Group 1, 2, and 3 on Pseudocode was 27 s, 44 s, and 31 s respectively. Group 1 took the least amount of time on average but still attained the highest number of correct answers on average.

ANOVA						
Source of variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between groups	7.43743E–07	2	3.71872E–07	8.799659143	0.000305682	3.089203013

(*continued*)

Table 4. (*continued*)

ANOVA						
Within groups	4.14146E–06	98	4.22598E–08			
Total	4.8852E–06	100				

Natural Language

See Tables 5 and 6.

Table 5. Average for correctness of problems for Group 1, 2, and 3 on Microsoft Excel was 0.962962963, 0.965811966, and 0.838095238 respectively, with Group 2 with the highest performance by a small margin compared to Group 1. Group 1 had the second set of problems in natural language, while Group 2 had their first set of problems, and Group 3 had their third set of problems in natural language.

ANOVA						
Source of variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between groups	0.366418034	2	0.183209017	5.984990277	0.003528941	3.089203013
Within groups	2.9999186	98	0.030611414			
Total	3.366336634	100				

Table 6. Average for the time taken for Group 1, 2, and 3 on Pseudocode was 34 s, 38 s, and 42 s respectively. Group 1 took the least amount of time on average but still attained a high amount of correct answers on average.

ANOVA						
Source of variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between groups	1.43218E–07	2	7.1609E–08	0.99835497	0.371798914	3.078819492
Within groups	7.88997E–06	110	7.1727E–08			
Total	8.03319E–06	112				

It was shown in the results that on average, students performed higher and took the shortest amount of time on natural language than pseudocode and Microsoft Excel.

On average, students performed lower and took the longest time on Microsoft Excel compared to natural language and pseudocode.

4.2 Section 2: Analysis Based on Group

Group 1
See Tables 7 and 8.

Table 7. Average for correctness of problems for Group 1 on the programs pseudocode, natural language, and Microsoft Excel was 0.974358974, 0.961538462, and 0.730769231 respectively, with the highest performance for pseudocode. Group 1 had the first set of problems in pseudocode, the second set of problems in natural language, and the third set of problems in Microsoft Excel.

ANOVA						
Source of variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between groups	0.977207977	2	0.488603989	13.44043887	1.02641E–05	3.118642128
Within groups	2.726495726	75	0.036353276			
Total	3.703703704	77				

Table 8. Average for the time taken for Group 1 on the programs pseudocode, natural language, and Microsoft Excel was 28 s, 27 s, and 35 s respectively. Although Microsoft Excel had a longer time duration, the accuracy rate was lower compared to pseudocode and natural language.

ANOVA						
Source of variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between groups	1.43495E–07	2	7.17473E–08	2.724328151	0.072084614	3.118642128
Within groups	1.97518E–06	75	2.63358E–08			
Total	2.11868E–06	77				