An evaluation of the usability of Notepad++ as an Environment for Java Programming at the National University of Samoa

Elisapeta Mauai, National University of Samoa

Edna Temese, National University of Samoa

Foilagi Faamausili, National University of Samoa

Tusipepa Malaga, National University of Samoa

Fiafaitupe Lafaele, National University of Samoa

Agnes Wong Soon, National University of Samoa

Tara Patu, National University of Samoa

Ioana Sinclair, National University of Samoa

Vensel Chan, National University of Samoa

Abstract

This paper presents a usability evaluation of Notepad++ (With Jdk Compiler) as an environment in learning Java Programming at The National University of Samoa (NUS) Foundation Level. The Foundation Computer Studies (HCS081) course is a servicing course to all NUS foundation programs in which introduces the programming component. The course instructors have observed that the common denominator of failure is largely due to their difficulty to get acquainted with the Integrated Development Environment (IDE) used. Students frequently struggled with learning and familiarizing themselves with the IDE used, losing focus on how to learn the logical sequence of a program, and the programming language itself. Consequently, this study aims to evaluate the usability of simple, open source tool; Notepad++ as the alternative programming environment, to teach and learn Java programming at the foundation level. The usability of the simple open source surveyed 178 Foundation students. The findings shows that there was high usability indicating the feasibility of Notepad as the preferred Java Development Kit (JDK) for teaching and learning Java in the Foundation level classes. In addition results from this study act to inform future research and best practice relevant in the NUS programming courses.

Keywords: Integrated Development Environment (IDE), Programming, Notepad++, Java Development Kit (JDK)

Introduction

It is often said that "Learning to program is at the heart of learning Computer Science..." (Malmi et al. 2005). Efforts are being made by universities in developing countries to ensure that their graduates are not left behind in the competitive global information society; thus have adopted computing degree programs in almost all their universities (Dasuki et al. 2015). The goal in teaching computer programming is to develop in students the capabilities required of a professional software developer. Unfortunately, learning to program is difficult (Kunkle and Allen 2016). Computer programming is a cognitively challenging subject and appears to be the most difficult aspect to master in dealing with computers (Malliarakis et al2014).

Numerous studies have been conducted on the difficulties of beginner programmers in learning as well as their having poor attitudes towards programming (Chan Mow 2012b; Flowers et al 2004; McIver, 2001; Thompson 2004). Research in computer education in developing countries has shown that one major cause of the low interest and high dropout rates in computer science degree programs is the difficulty and high failure rate in learning programming (Sarpong et al 2013). To this end, measures to support the teaching and learning of students enrolled in programming courses are critical to the discipline.

In general, the high failure of students who first take computer programming as a core course or a minor in their respective programs can be influenced by various factors. Such factors include lack of problem-solving and analytical skills; logical and reasoning; programming planning; algorithmic skills; pedagogical and teaching methodologies; conceptual difficulty of various elements of the curriculum; mathematics and previous academic backgrounds; cognitive factors, personality and learning styles; and the complexity and poor design of programming environments used (Kölling et al 2003). Since programming is the basic skill required of computer programmers, the negative impact of these factors may have harmful consequences on the learners' attitude towards the field, and often discourage them to take advanced programming courses. On the other hand, results of a study at NUS (Mauai and Temese 2012) indicated that prior knowledge in English, Mathematics and Computing are strong predictors of performance in Foundation Computer studies.

At the National University of Samoa, students who are enrolled in computing as a major are expected to undertake programming along with other computing subjects in the areas of applications, networking, hardware, and information management. At the Foundation level, HCS081 (Introduction to Computer Studies) is an introductory computing course which is compulsory in almost all programmes. In this introductory course, students learn the basic hardware/software components of computers, file management utilities, office applications such as word processing, spreadsheets, databases, as well as basic programming. The programming component of the course uses Java as the programming language with the JBuilder software as the Integrated Development Environment (IDE). The programming component covers an introduction to programming concepts such as classes, objects, variables, methods, and condition executions using object-oriented programming (OOP) approach.

In the past few years, it has been observed that most of the students failed the course, due to their poor performances in the programming component. Lecturers and tutors for HCS081 discovered that the majority of students were frequently struggling in trying to learn and familiarize themselves with the programming environment used, losing focus on how to learn the logical sequence of a program, and the programming language itself.

Previous studies showed evidence that IDEs are too often functionality-oriented and difficult to learn, use, and master at the introductory level (Kölling et al, 2003). Such challenges and difficulties result in lower student retention and difficulty in passing programming courses. Learning how to use menu-driven programming environments such as JBuilder may take some time for the average student to comprehend, and generally ended up learning the environment, but not grasping the essential concepts of programming (Horstmann 2019).

In Samoa, three of the few studies on computer programming were conducted (Chan Mow 2006; Chan Mow 2008, Chan Mow 2012b) at NUS. The first study by Chan Mow (2006) evaluated the effectiveness of CABLE (Cognitive Apprenticeship Based Learning Environment) as a teaching environment for programming within a university context. The second study (Chan Mow 2008) investigated issues in the instruction of computer programming. The third (Chan Mow 2012b) aimed to investigate specifically the types of errors students make in programming; however, it did not identify areas/topics of difficulty, deficiencies in learning support, or recommendations for improving the learning environment from the students' point of view.

Furthermore, an investigative study on NUS Computer Programming Students' Perceptions of their computer programming experience by Chan Mow (2012a) indicated that despite their high motivation on their programming experiences, students found programming difficult, and also involved a lot of work and readings. The study also identified the fact that the students' level of engagement is inadequate, which seemed to indicate that students did not feel sufficiently engaged in their programming environment.

Research Objectives

This study was initiated to investigate various ways to solve the outlined problem. Moreover, an analysis of issues and challenges could inform teaching practice in terms of teaching and learning approaches specifically in selecting appropriate learning tools.

In this study, we explored the usability of some of the simple tools that are freely available, that could be used as alternative environments in learning Java programming. The integrated environment, (as we would like to call it), includes the Notepad++ code editor application, and the Java Development Kit (JDK) compiler. The Notepad++, was used by students to compose and edit their computer programs. The programs were then to be compiled and then executed by the JDK or the compiler using the MS-DOS command-based shell. The JDK translates the Java Language into the machine code. The JDK comprises of the Java Virtual Machine (JVM) compiler that does the compilation process. These two separate applications made up the integrated environment that our study evaluated.

The study was implemented at NUS for the first semester in 2019 to all the students who were enrolled in HCS081. The evaluation included a series of basic programing tasks that was performed by students throughout a period of four weeks for that semester. Throughout the four-week timeframe, students initially spent some time learning and familiarizing themselves with the Notepad++ and JDK environment. Hence the research questions are:

1. What are the students' perceptions in terms of skill- level and usage of programming languages?
2. How usable is Notepad++ with the JDK Compiler as tools to learn basic Java programming in terms of usability, effectiveness, and satisfaction?

Accordingly, the main goal of this research is to explore the possibility of using Notepad++ and JDK compiler as an alternative programming environment to learn Java programming at introductory level by evaluating its usability. It also aimed to Identify of issues and challenges in learning programming using our proposed learning environment, to provide valuable feedback that can be used to revise and improve

the curriculum, teaching strategies and learning support specifically at the Foundation level. Findings from the study will also recognize if using such integrated environment for learning Java programming at the introductory level is usable or not.

Usability Criteria

Usability, as defined in International Organization for Standardization (Guidance on Usability 2018), is: "the extent to which a product can be used by specified users to achieve goals with effectiveness, efficiency and satisfaction in a specified context of use." An elaboration of this in terms of software usability is: "software is usable when the user can achieve their task (effectiveness), with a minimum amount of resources (efficiency), and that the system is pleasant to use (satisfaction)". In this context, the more usable a system is, the more possible it is for a user to continue using the system efficiently over a long period of time.

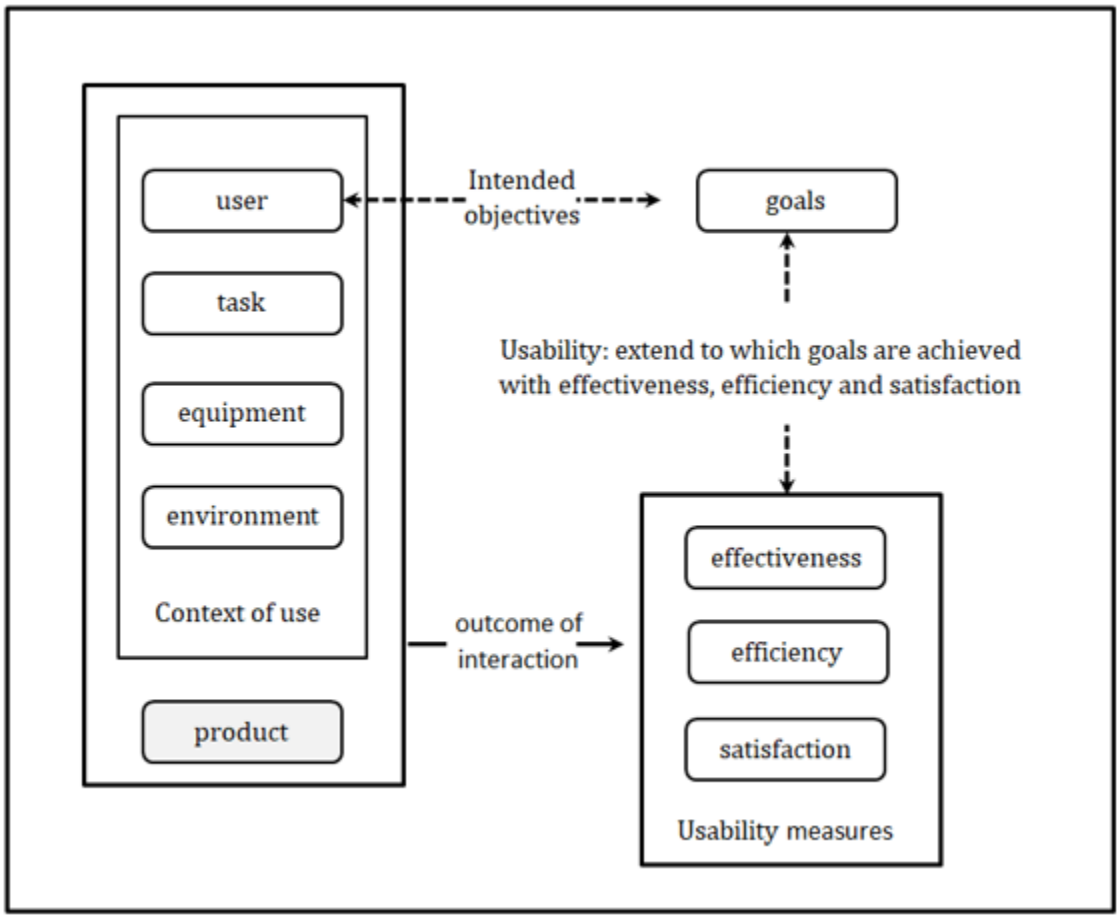
Historically, the concept of usability has been defined in multiple ways, but usually on one of the following bases:

i. Semantics: in this case, usability of the syntax language describes a meaning and is easy for user to understand and processes.
ii. Features: here, usability is equated to the presence or absence of certain features in the user interface such as Windows, Icons, Menus or Pointing devices.
iii. Operations: where the term is defined in terms of performance and affective levels manifest by users for certain task and environmental scenarios.

This study takes on the operations approach. Shackel (1991), as the major developer of the operational approach defined usability as the artefact's capability, in human functional terms, to be used easily, effectively and satisfactorily by specific users, performing specific tasks, in specific environments. The essence of the operation approach definition is that it explicitly places usability at the level of the interaction between users and the artefact. Usability therefore refers not to a set of interface features, but to a context-dependent measure of human-computer interaction. The operation approach is interrelated to user-based (Dillon 2001). User-based or testing an application with a sample of users performing a set of pre-determined tasks is generally considered to yield the most reliable and valid estimate of an application's usability.

The Usability: definitions and concepts (2015) explains how usability can be specified and evaluated in terms of user experiences with the system and how satisfied they are in using the system. It also emphasizes that usability is dependent on the context of use and that the level of usability achieved will depend on the specific circumstances in which a product is used. The context of use consists of the users, tasks, equipment (hardware, software and materials), and the physical and organizational environments which may influence the usability of a product (see Figure 1 below).

Figure 1: Usability Framework



The critical point in terms of usability of a programming environment is the extent to which a user can transfer his or her existing knowledge to the tool. In terms of programming environments, students should not have to invest a large amount of time obtaining, installing, and learning to use a tool before they can begin to learn programming (Van-Haaster and Hagan 2004).

In the IEEE's Software Engineering Body of Knowledge (SWEBOK), (Bourque and Fairley, 2004) specified that one of the goals that a software development environment (which includes IDEs) should meet is to reduce the cognitive load on the developer in order to free the developer to concentrate on the creative aspects of the process. Unfortunately, there is evidence that these goals are often not met in practice and that poor IDE usability is a culprit. IDEs are described by many developers as difficult to learn and use (Seffah and Rilling 2001). Printed documentation for developers, online help, and related training materials are often presented in language that is precise but esoteric and difficult to understand (Seffah 1999).

In a study on the usability of Eclipse for novice programmers, it was found that Eclipse has many powerful features that can assist both expert and novice programmers and Java instructors (Storey et al 2003). However, they believed that Eclipse can be improved to offer enhanced support to novice programmers as well as instructors. Specifically, the Eclipse user interface may be overly complex and that

this complexity could pose a barrier to learning. Furthermore, they have identified that in terms of application development, it is hard to eliminate the problem of IDE dependence. When a student's productivity becomes tied to a certain IDE, it is a serious liability (McIver 2002). A shift in school curriculum standards or a change of programming environment from one tool to another can precipitously affect students' productivity and their knowledge of the previous environment used. At the introductory level, by using an open source application such as Emacs or Notepad++, we are not building a dependency on any IDE's specific features, and students can easily reuse their knowledge and adapt it to a new learning environment.

A lot of undergraduate courses in universities today are moving towards teaching Object Oriented Programming (OOP) as the first programming subject introduced in first year students. The abstract concepts involved in understanding OOP and the lack of appropriate teaching aids make OOP hard to teach in the classroom (Kölling et al 2003). In a recent study by Bettini and Crescenzi (2015), they introduced Java--. Java -- or Java decrement is an application that allows students to learn programming by using smaller version of Java without object-oriented features. Using this approach, it combines the "structured programming before object oriented programming", hence, students can focus on the basic programming concepts without being distracted by complex constructs. They further stressed that when students are still new to Java and are just beginning to explore the language, keeping things simple with a text editor and a compiler helps them focus on the basics. At this stage of learning, the student wants to spend most of his/her time becoming intimate with the nuances and fundamental concepts of the language. Once they've become comfortable with the basics, then they can use an IDE to help them with the next phase of learning.

Methodology

The study is quantitative in nature through the use of survey questionnaires to collect data. The survey instrument consisted of two parts:

i. General Demographics and a Pre Assessment Questionnaire
ii. Students Perceptions/Attitudes towards the efficiency, effectiveness and satisfaction of the Learning Environment

Population and Sample

The target population included all students enrolled in HCS081 (Foundation Computer Studies) in Semester one (1), and Semester two (2), in 2019. The time frame of the user-study was four (4) weeks for each semester. This coincided with the last four weeks of each semester which were allocated to the Java programming component of the HCS081 course. Throughout the four weeks' time-frame, students were exposed to, and experimented with Notepad++ and JDK through tutorials and practical programming activities.

Procedures

The evaluation questionnaire which was given out at the end of the programming component evaluated the usability of the environment based on user's perceptions after they learnt and experimented with the environment.

The allocated tasks and programming activities covered the basic concepts of Java programming focusing on object-oriented approach. This included simple variable declaration, assignment statements, simple arithmetic calculations, conditionals, and output statements.

The questionnaires developed gathered data measuring the dependent variables of students' perceived level of satisfaction, effectiveness, and efficiency, in relation to the independent variable which was the learning environment. In addition to the questionnaires, students' performance in programming tasks were also recorded and evaluated, for further analysis but is not included in this study's findings.

Data Analysis

As previously stated, the data that was collected in this study were in the form of answered questionnaire items, falling into the interval scale of measurement. Data analysis was carried out using a combination of analysis software including MS Excel and SPSS.

Statistical analysis was conducted using SPSS with the data primarily being descriptive statistics. Reliability analysis was conducted using alpha-Cronbach indexes.

Results and Discussion

The results of the analysis are reported in the sections in the order within which they are located in the questionnaire. In the case of multiple response items, the findings are reported as percentages or proportion of the total sample of 178.

General Demographics and a Pre Assessment Questionnaire

Nature of Study Participants

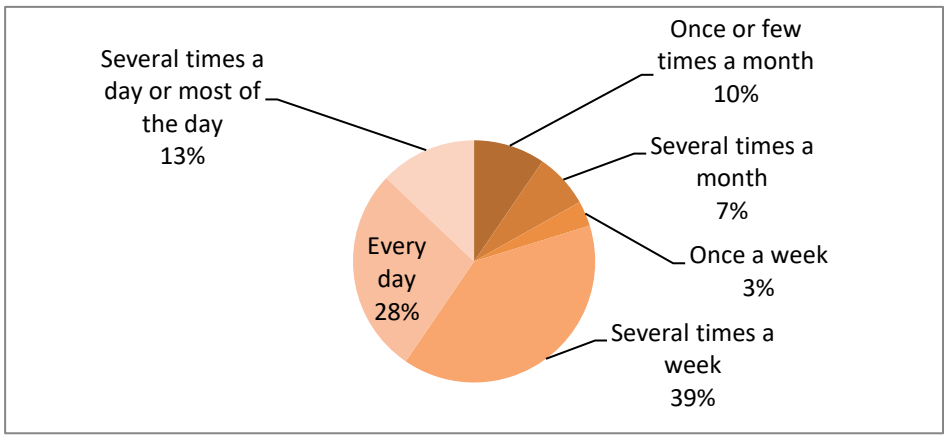
There were 178 total survey respondents enrolled in the various foundation programs with 82 from Foundation Certificate in General (FCG), 62 from the Foundation Certificate in Commerce (FCC) 15 from the Foundation Certificate in Science (FCS), 12 from the Foundation Certificate of Art (FCA), one from Certificate of Commerce Study (CSS) and ten from Diploma of Maths (DipMaths). Most respondents were in the age range of 16 to 19 years old whist 12 were in the age range of 20-23. In terms of gender, respondents were predominantly female with 26 percent male and 74 percent female.

Users Computer Skills

Asked: How often do you use a computer? Responses indicated that 39 percent of the participants use computer several times a week, 28 percent uses computers every day and 13 percent use computers several times a day and most of the day. Ten (10) percent claim they use computer once or few times a

month, seven (7) percent use computer several times a month whist three (3) percent use computer once a week.

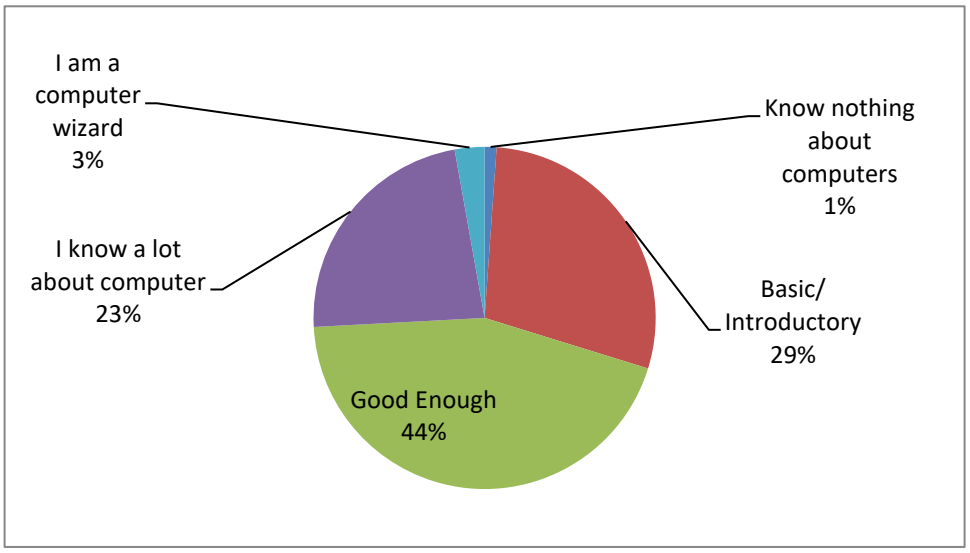
Figure 2: Computer use



Self-evaluation of computer knowledge and skills.

An evaluation of computer knowledge and skills indicated that 44 percent of the respondents rate their general level of computer knowledge and skills as good enough; 23 percent thought they are at the basic or introductory level while three percent indicated that they are computer wizards. On the other hand, 29 percent indicated they have basic and introductory computer knowledge and skills and one percent claimed that they knew nothing about computers.

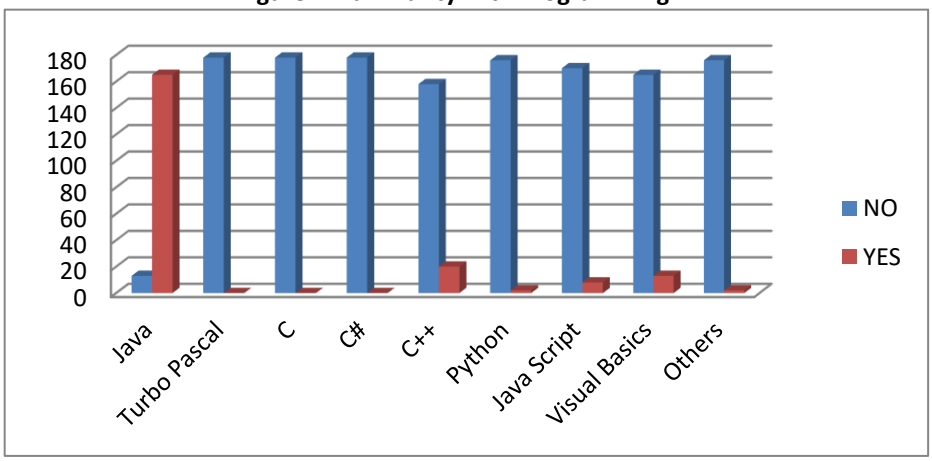
Figure 3: General level of computer knowledge and skills



Familiarity with programming languages.

In terms of familiarity with programming languages, 93 percent of the participants are familiar with Java, 11 percent know C++, seven (7) percent said they are familiar with Visual Basics, four (4) percent indicated familiarity with Java Script language, and one (1) percent indicated they were familiar with Python programming language. As alluded to earlier, the Java language has been taught in Secondary schools as well as at University level, thus many who took computer courses have background knowledge in the Java content.

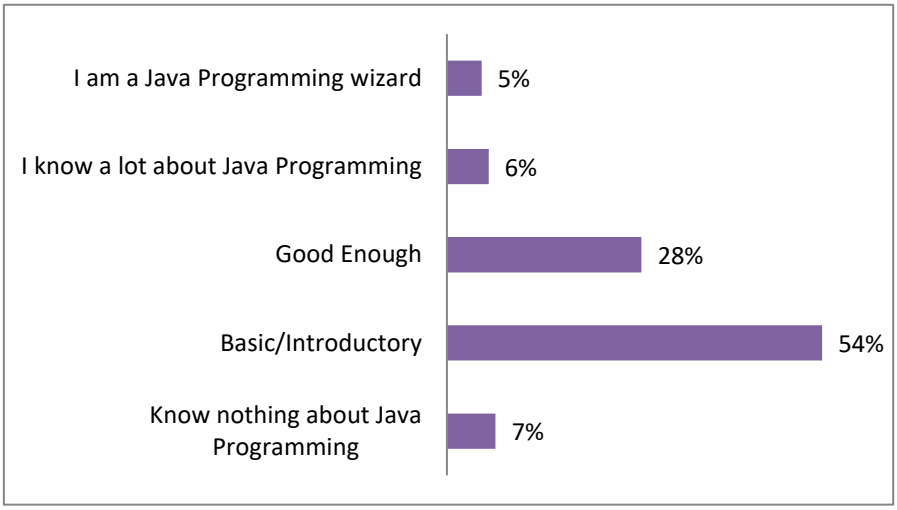
Figure 4: Familiarity with Programming



Self-assessment of knowledge of computer programming in Java

A self-assessment of the general level of computer programming in Java, revealed that five (5) percent of the participants claimed they are Java programming wizards, six (6) percent indicated they know a lot about Java programming, and 28 percent of the respondents claim they rate their general level of Java as good enough. Majority of respondents claim having a basic or introductory level of Java background (54 percent) whist seven (7) percent indicated they knew nothing about Java programming. The results indicated that the majority perceived themselves as knowledgeable on the basics of Java programming.

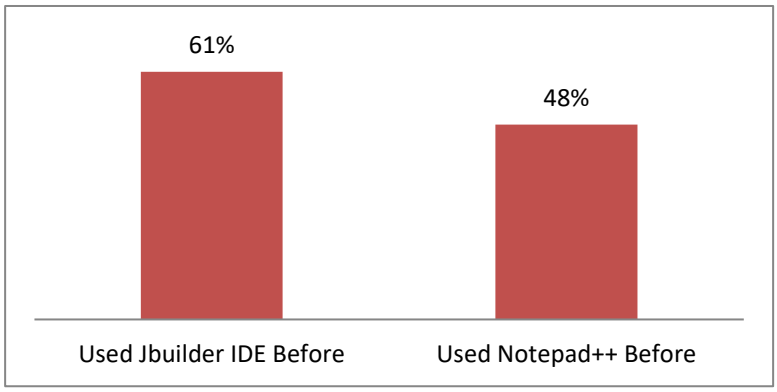
Figure 5: General level of Java programming



Prior use of JBuilder IDE and Notepad++

Respondents were asked whether they had ever used the JBuilder IDE and the Notepad++ before. Respondents indicated that 61 percent used JBuilder IDE before while 48 percent of the respondents used Notepad++ before. This is to be expected as the JBuilder IDE is used in the Secondary schools and the Foundation Java Programming language and the current cohort are using the Notepad++.

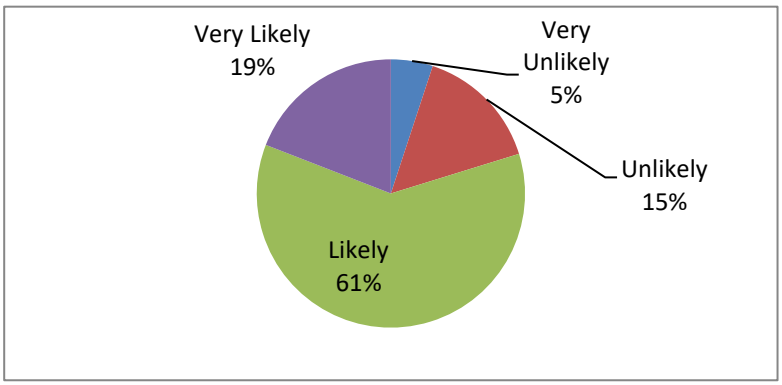
Figure 6: Use of JBuilder IDE and Notepad before



Self-assessment of the likelihood the student take a programming course

The probe on the likelihood that respondents were to take a programming course, showed that the majority (61 percent) were likely to take a programming course with 19 percent indicating that they were very likely to take programming course. 15 percent of the participants were unlikely to consider being interested, with 5percent claiming as very unlikely to take a programming course at all in the future. Such responses are useful in guiding decisions on strategies to boost the interest of students in taking and learning programming languages.

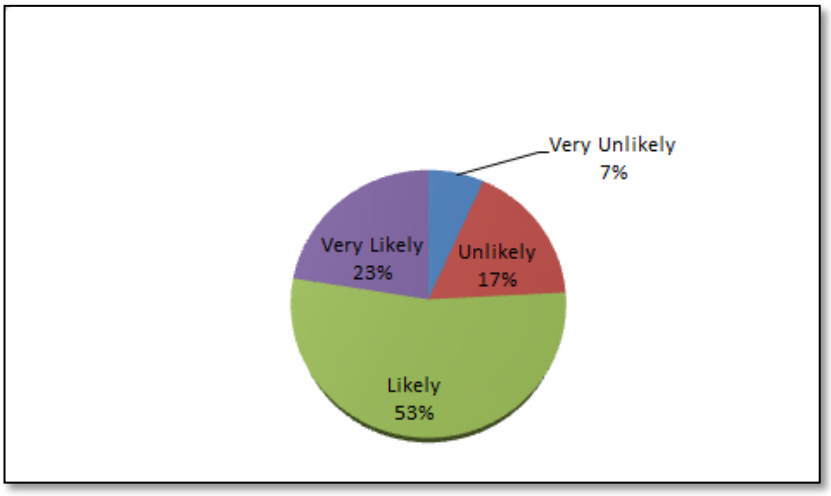
Figure 7: Likely to take a programming course



Future career aspirations for jobs that involves computer programming or software development

The most common responses were likely (53 percent) followed by very likely (23 percent), unlikely (17 percent) and with seven (7) percent of respondents indicating that they were very unlikely to pursue a career involving Computing Programming.

Figure 8: Likely to Pursue a computing programming related job



Students' perceptions of the efficiency, effectiveness and satisfaction of the Learning Environment

The responses on student perceptions on usability evaluation (PUEQ) show how students rated the usability of the Notepad++ environment after they have learnt and experimented with that environment. PUEQ was developed with 6 question items grouped under each category of effectiveness, efficiency and user satisfaction consecutively. Responses were rated using five options on a Likert scale ranging from Strongly Disagree (1), Disagree (2) Neutral (3) Agree (4) Strongly Agree (5). Table 1 shows 3 sample questions under these categories.

Table 1: Sample questionnaire items of effectiveness, efficiency and user satisfaction

			1	2	3	4	5	
Q1	**PQ1. I can effectively complete my work using this environment**	strongly disagree	○	○	○	○	○	strongly agree
Q10	**PQ10.I was able to efficiently complete my work using this environment**	strongly disagree	○	○	○	○	○	strongly agree
Q16	**PQ16.My general level of satisfaction with this learning environment was**	strongly disagree	○	○	○	○	○	strongly agree

Reliability is the consistency of test results obtained when a test instrument is applied to different testing environments and yield the same results. The PUEQ questionnaire was tested through evaluating Cronbach's alpha minimum reliability value of 0.07 on 178 respondents. If the table value is > 0.07 it indicates a high level of reliability of the instrument and the items tested. Derisma (2020) tested the reliability on a usability analysis using the System Usability Scale (SUS) of the Codesaya learning site and confirmed that the table value of 0.968 was greater than 0.07. Therefore, the SUS questionnaire is highly reliable.

Table 2: Reliability statistics

Measures	Items	Mean	Std. Deviation	Cronbach's Alpha
Effectiveness	6	3.80	0.84	0.899
Efficiency	6	3.77	0.85	0.920
Satisfaction	6	3.70	0.84	0.897

From Table 2, effectiveness, efficiency and user satisfaction have Cronbach Alpha table values of 0.899, 0.920 and 0.897 consecutively. All these table values are greater than 0.07. Table 3 shows the overall value of Cronbach's Alpha as 0.962 for all the 3 items thus confirms that the PUEQ questionnaire is highly reliable.

Table 3: Overall reliability statistics

Cronbach's Alpha	N of Items
.962	3

For survey items probing the effectiveness of Notepad++, students mean ratings per item ranged between 3.66 to 3.92 as shown in Table 4. This shows that the majority of students have strong positive perceptions of Notepad++ as an effective Integrated Development Environment for learning and completing Java tasks. With student perceptions yielding high means on probes on useful error messages, effective completion of work and shown interests to use the environment frequently, this points to strong positive perceptions on the effectiveness of Notepad++.

Table 4: Comparison for Means

Measures	Items	Item Mean	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Effectiveness	1. I can effectively complete my work using this environment	3.88	2.4	2.4	3.1	3.9	4.6
	2. I believe I became productive quickly using this learning environment	3.80	1.4	2.6	3.2	4	4.6
	3. The environment gives error messages that clearly tell me how to fix problems	3.92	2.3	2.5	3.3	3.8	4.4
	4. I found the various functions in this environment were well integrated	3.70	1.9	2.9	3.3	3.9	4.8
	5. I was satisfied with the environment's capabilities in facilitating the completion of the allocated task in this session	3.66	1.7	2.8	3.3	4	4.7
	6. I think that I would like to use this environment frequently	3.86	1.4	2.9	3.2	3.9	4.6

Analysis of survey items which evaluated the efficiency of Notepad++, in Table 5 showed means for item responses ranged from 3.67 to 3.94 which again indicated fairly positive responses in terms of the efficiency of Notepad++. Thus these results indicated how easy it was to use this environment, the level of efficiency and how quickly it took to complete their work and satisfaction with the amount of time it took to complete tasks using the Notepad++ environment.

Table 5: Mean of efficiency

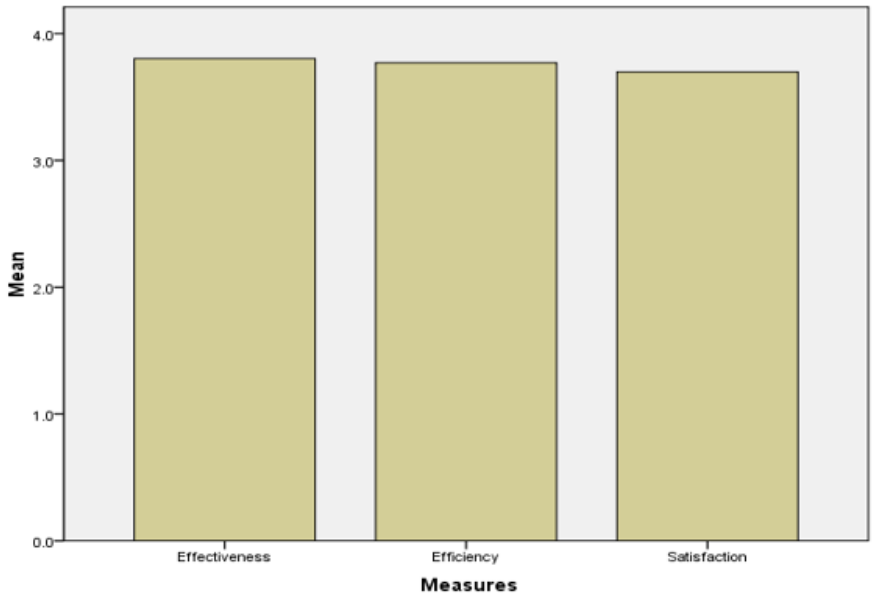
Measures	Items	Item Mean	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Efficiency	7. Overall I was satisfied with how easy it was to use this environment	3.67	1.9	2.6	3.4	4	4.7
	8. I was able to complete my work quickly using the environment	3.76	1.5	2.6	3.3	3.9	4.7
	9. I felt comfortable using this learning environment	3.94	1.2	2.4	3.1	3.8	4.6
	10. I was able to efficiently complete my work using this environment	3.69	1.2	2.5	3.3	4	4.7
	11. Overall, I was satisfied with the ease of completing the task in this learning environment	3.81	1.7	2.3	3.2	4	4.6
	12. Overall, I was satisfied with the amount of time it took me to complete the tasks in this scenario	3.74	2.3	2.6	3.3	4	4.6

An examination of user satisfaction of Notepad++, the respondents mean rating per item ranged from 3.58 to 3.87 as given in Table 6. This generally indicates that student views of satisfaction are fairly positive with respect to environment for teaching and learning Java Programming at the Foundation Level, they were fairly confident to use this environment and felt quite engaged in using Notepad++ during the task activity.

Table 6: Means of user satisfaction

Measures	Items	Item Mean	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Satisfaction	13. It was easy to learn to use this learning environment	3.67	2.1	2.4	3.3	3.9	4.6
	14. I felt engaged with this learning environment during the task activity	3.71	2.2	2.6	3.2	3.8	4.6
	15. I was satisfied with how easy it was to use this learning environment	3.64	1.9	2.7	3.2	4	4.6
	16. My general level of satisfaction with this learning environment was	3.58	1.9	2.6	3.2	4	4.7
	17. Please rate your level of confidence in using this learning environment	3.71	1.1	2.6	3.2	3.9	4.6
	18. I would recommend using this environment in teaching and learning Java Programming at the Foundation Level	3.87	1.8	2.7	3.2	3.8	4.3

Figure 9: Overall perception of Notepad++



An overall analysis of the data items was done using an interpretation of mean results of each individual item as discussed. The mean result on the Likert scale is 3. Graph 8 presents a comparison of means of all items in the 3 groups of effectiveness, efficiency and user satisfaction. The means of items in all 3 categories are all >3. This is an indication that the responses and perceptions of the research participants are highly positive towards supporting the effectiveness, efficiency and satisfaction of users when using Notepad++ as an alternative learning tool for computer programming.

The validity of the research instrument is confirmed by the extent to which the items are internally correlated and relevant towards what the items are measuring. The validity of this PUEQ questionnaire was tested on a total of 178 participants and using SPSS to test the correlation of the three aspects of effectiveness, efficiency and user satisfaction using the Pearson correlation value of a perfect positive correlation as indicated by r at the 0.01level. Table 8 shows the correlation of each measure.

Table 7: Correlation of each measure

Measures		Effectiveness	Efficiency	Satisfaction
Effectiveness	Pearson Correlation	1	.899**	.905**
	Sig. (2-tailed)		.000	.000
	N	178	178	178
Efficiency	Pearson Correlation	.899**	1	.878**
	Sig. (2-tailed)	.000		.000
	N	178	178	178
Satisfaction	Pearson Correlation	.905**	.878**	1
	Sig. (2-tailed)	.000	.000	
	N	178	178	178
**. Correlation is significant at the 0.01 level (2-tailed).				

The first row shows a strong positive correlation of effectiveness in line with efficiency (.899) and user satisfaction (.905). The second row shows another strong positive correlation among efficiency in line with effectiveness (.899) and user satisfaction (.878). The third row presents another strong positive correlation between user satisfaction as the measured factor along with effectiveness (.905) and efficiency (.878). Overall Table 1 shows that the correlation is significant at the 0.01 level.

Conclusion and Recommendations

This is the first review focusing exclusively on the Foundation students' perception by exploring the possibility of using Notepad++ and JDK compiler as an alternative programming environment to learn Java programming at the introduction level. Based on the results of the analysis performed, although the sample population are novice users, students' perception in terms of skills and usage of Programming languages was high. Nevertheless, the study shows the Notepad++ as the code editor application and the JDK compiler as a tool to learn basic Java Programming has a high usability, indicating the feasibility of Notepad++ in terms of effectiveness, efficiency, and satisfaction. This confirms a recommendation to use Notepad++ and JDK compiler as an alternative programming environment to teach and learn Java programming at Foundation level. While not necessarily a limitation in this study, the idea of using

Notepad++ eliminates the need for students to go through a series of complex of the existing default steps in Java allowing them to focus on the programming concepts.

Acknowledgement

The work described in this study was initiated by Mr Hobert Sasa and the study was later transferred to the research team. We would like to thank the UREC for providing the research grant and Professor Ioana Chan Mow for mentoring the research team and provided a high level feedback on the study. The completion could not have been possible without the participation and assistance of the HCS081 students enrolled in 2019.

References

Bettini, L, & Crescenzi, P 2015, 'Java-meets eclipse: An IDE for teaching Java following the object-later approach', In *2015 10th International Joint Conference on Software Technologies (ICSOFT)*, IEEE, 1-12

Bourque, P., & Fairley, R. (2004) 'Software Engineering Body of Knowledge' IEEE Computer society.

Chan Mow, ITV 2006, 'The effectiveness of a cognitive apprenticeship-based learning environment (cable) in teaching computer programming' Ph.D. thesis

Chan Mow, ITV 2008, 'Issues and difficulties in teaching novice computer programming.' in Iskander, M (eds) *In Innovative techniques in instruction technology, e-learning, e-assessment, and education conference*, Springer, Dordrecht, pp. 199-204.

Chan Mow, ITV 2012a, 'An Investigative Study on NUS Computer programming Students' Perceptions of their computer programming experience: some preliminary findings', *International Journal of Information,* 1(1).

Chan Mow, ITV (2012b). Analyses of student programming errors in Java programming courses. *Journal of Emerging Trends in Computing and Information Sciences*, *3*(5), 739-749.

Dasuki, S, Ogedebe, P, Kanya, R, Ndume, H, & Makinde, J 2015 'Evaluating the implementation of international computing curricular in African universities: A design-reality gap approach', *International Journal of Education and Development using ICT*, 11(1).

Derisma, D 2020, *The Usability Analysis Online Learning Site for Supporting Computer programming Course Using System Usability Scale (SUS) in a University,* International Association of Online Engineering viewed June 7, 2021 from https://www.learntechlib.org/p/217827/.

Dillon, A 2001, 'The evaluation of software usability', *Encyclopedia of Human Factors and Ergonomics* London: Taylor and Francis.

Flowers, T, Carver, CA, & Jackson, J 2004, October. Empowering students and building confidence in novice programmers through Gauntlet. In 34th Annual Frontiers in Education, 2004. FIE 2004. (pp. T3H-10). IEEE.

Horstmann, CS 2019, *Big Java: Early Objects.* John Wiley & Sons.

Guidance on Usability, 2018, *Ergonomic requirements for office work with visual display terminals (VDTs)*, Part 11(ISO 9241-11).

Usability: definitions and concepts, 2015, *Ergonomics of human-system interaction,* Part 11(ISO 9241-11).

Kölling, M, Quig, B, Patterson, A, & Rosenberg, J 2003 'The Blue J system and its pedagogy', *Computer Science Education,* 13(4), 249-268.

Kunkle, WM, & Allen, RB 2016, 'The impact of different teaching approaches and languages on student learning of introductory programming concepts', *ACM Transactions on Computing Education* (TOCE), 16(1), 1-26.

Malliarakis, C, Satratzemi, M, & Xinogalos, S 2014, Integrating learning analytics in an educational MMORPG for computer programming. IEEE 14th international conference on advanced learning technologies. IEEE, pp. 233-237.

Malmi, L, Karavirta, V, Korhonen, A, & Nikander, J 2005, 'Experiences on automatically assessed algorithm simulation exercises with different resubmission policies', *Journal on Educational Resources in Computing (JERIC),* 5(3), 7-es.

Mauai, E, & Temese, E 2012, 'Exploratory study on factors that Impact/Influence Success and Failure of students in the Foundation Computer Studies Course at the National University of Samoa', *Journal of Emerging Trends in Computing and Information Sciences,* 3(5), 767-773.

McIver, LK 2001, 'Syntactic and semantic issues in introductory programming education', PhD thesis, Monash University.

McIver, L. (2002, June). Evaluating Languages and Environments for Novice Programmers. In PPIG (p. 10).

Sarpong, KAM, Arthur, JK, & Amoako, PYO 2013, 'Causes of failure of students in computer programming courses: The teacher-learner Perspective', *International Journal of Computer Applications*, *77*(12).

Seffah, A. (1999). Training developers in critical skills. *IEEE software*, *16*(3), 66-70.

Seffah, A, & Rilling, J 2001, Investigating the relationship between usability and conceptual gaps for human-centric CASE tools. In *Proceedings IEEE Symposia on Human-Centric Computing Languages and Environments (Cat. No. 01TH8587)* (pp. 226-231). IEEE.

Shackel, B., & Richardson, S. J. (Eds.). (1991). Human factors for informatics usability. Cambridge university press.

Storey, MA, Damian, D, Michaud, J, Myers, D, Mindel, M, German, D, & Hargreaves, E 2003, Improving the usability of Eclipse for novice programmers. In *Proceedings of the 2003 OOPSLA workshop on eclipse technology eXchange* (pp. 35-39).

Thompson, SM 2004, *An Exploratory Study of Novice Programming Errors and Experiences.* Unpublished thesis (MSc), Victoria University.

Van-Haaster, K., & Hagan, D. (2004). Teaching and Learning with BlueJ: an Evaluation of a Pedagogical Tool. *Issues in Informing Science & Information Technology*, *1*.