ISSN 2319 - 7595
Volume 1, No.1, September - October 2012
International Journal of Information Systems and Computer Sciences
Available Online at http://warse.org/pdfs/ijiscs02112012.pdf

Investigative Study on NUS Computer programming Students' Perceptions of their computer programming experience: some preliminary findings

Ioana Tuugalei Vaai Chan Mow
[1]National University of Samoa, Samoa, i.chanmow@nus.edu.ws

ABSTRACT

This paper describes some aspects of a study aimed at investigating student perceptions' of their computer programming experiences at the National University of Samoa (NUS). The objectives were to: i)gauge student opinions on interest and experience, level of mental engagement, motivation, with respect to their programming skills and experience ;ii) identify perceived areas/topics of difficulty in the programming curriculum and iii) identify perceived deficiencies in learning support and the learning environment. Students in the 3 programming classes were given questionnaires to complete. The results in this paper is limited to responses from one class, HCS286 and focuses on findings on gauging student opinions on interest, experience, level of mental engagement and motivation with respect to their programming skills and experience. Findings from the current study provide useful feedback which can be used in improving the student learning environment for such a cognitively challenging subject. These preliminary findings will be further elaborated upon analyses of other aspects of the study.

Keywords : Java programming, Learning Support, Motivation, Student Perceptions.

1. INTRODUCTION

Computer programming is a cognitively challenging subject and appears to be the most difficult aspect to master in dealing with computers [1] and computer science educators are growing increasingly concerned over the lack of programming comprehension of novice computer science students.

At the National University of Samoa (NUS) Computer programming courses are taught as part of the undergraduate programs in Computing (certificate, diploma and bachelors). Except for one course in Visual basic, the rest of the courses are in Java programming. There are 4 Java programming one-semester courses in the undergraduate programming strand: HCS181, HCS281, HCS286 and HCS381. The prerequisite to HCS181 is HCS081 which contains a section on Java programming. The HCS081 Java section provides an introduction to programming concepts, introduces the integrated developer environment (IDE) JBuilder (java program editor, compiler and debugger), teaches simple java applications using projects, classes, methods and attributes, basic sequential and conditional executions. HCS181 provides an introduction to programming concepts, link to software development, introduces the integrated developer environment (IDE) JBuilder (java program editor, compiler and debugger), UML modeling, teaches simple java applications using projects, classes, methods and attributes, arrays, sequential, conditional and loop executions. HCS281 is the sequel to HCS181 which includes conditional and loop executions, exception handling, reading/writing from keyboard, reading/writing text files. HCS286 which is the sequel to HCS281 continues with exception handling, reading/writing from keyboard, reading/writing text files, access levels and then introduces inheritance, static objects, polymorphism, overriding, arraylists, linked lists, queues and stacks, hash tables, sorting, searching and binary trees. HCS381 the final Java course extends concepts learnt in HCS286 and students also focus on programming projects and developing applications.

With the cognitively challenging nature of computer programming it is important that an investigation be conducted on the perceptions of students of their programming learning experience. Such an analysis will identify areas or topics which students find or perceive as challenging, and issues and problems in the learning environment. According to [2] a learning environment consists of 3 components: the external environment which consists of the physical learning environment and learning activities, sensors, and the internal environment which includes emotions in learning, stimulating intelligence and understanding ways of learning.

For the current study, it is conceptualized that an effective learning environment is one which provides students i) with intrinsic satisfaction in the form of improved interest, enhanced motivation and levels of engagement, and ii) extrinsic support in the form of effective learning support and learning resources. Hence the factors investigated in the proposed study include motivation, level of engagement, learning support and resources. Motivation is a factor which is key to students learning [3]. Students' motivation to study directly influences their attitude to their work [3] – [4]. According to Ormrod [5], in order to motivate students, teachers should enhance students' expectations of success by providing the necessary resources, support, and strategies. According to constructivist approaches, active engagement of the learner is required for learning to take place [3]. In the proposed study, student engagement will be assessed on the level of students' investment in, and their emotional reactions

to, the learning tasks (e.g., high levels of interest or positive attitudes towards in the learning tasks). Engagement levels have also been found to relate positively to students' confidence and self-efficacy for achieving specific learning outcomes [6].

The research question for the study is:

"What are student perceptions their computer programming courses at the National University of Samoa (NUS)?"

Specifically, the objectives of the study are to:

- Gauge student opinions on interest and experience, level of mental engagement, motivation, with respect to their programming skills and experience.
- Identify perceived areas/topics of difficulty in the programming curriculum
- Identify perceived deficiencies in learning support and the learning environment.

Gauging students' level of interest, experience, level of engagement and motivation are important indicators of the effectiveness of the learning environment. Identification of issues and challenges in learning programming will provide valuable feedback that can be used to revise and improve the curriculum, teaching strategies and learning support. Identified deficiencies in learning support can be used to improve the learning environment. Findings from the study will identify the most problematic areas which lecturers need to bring attention to. This will help in allocation of time for topic coverage devoting the most time and attention to topics and areas identified by students as being the most difficult. It may even be feasible to use such information to identify at risk students at an early stage and provide targeted intervention.

2. LITERATURE REVIEW

A considerable number of studies have been conducted on the difficulties of novice programmers in learning programming [7] – [10]. The rationale has been that if we can understand the process of learning a first programming language, then we can create more effective learning environments. Furthermore, an analysis of issues and challenges could inform teaching practice in terms of teaching approaches and time allocations of topics.

In Samoa, two studies on computer programming were conducted by the principal researcher [11] – [12] at NUS. The first study was conducted by [11] Chan Mow in 2006 in which over a three-year period, an instructional program for teaching computer programming at university level was developed, and referred to as CABLE (Cognitive Apprenticeship Based Learning Environment).

From this investigation, a learning environment CABLE was designed which made use of cognitive apprenticeship, collaborative learning, meta-cognition, and technologies through the use of tele-apprenticeship and online or computer-mediated communication (CMC). From the field trials of CABLE, it was found that: (a) CABLE provided a viable instructional model which could be introduced into the normal conduct of university programming courses, (b) students exposed to CABLE evidenced increased achievement on Java programming scores relative to those taught in the traditional mode, (c) there were no differences in student attitudes towards the learning environment, between students taught with CABLE, and those taught in the traditional university mode, and (d) students taught under CABLE reported higher levels of mental engagement when compared to students taught via traditional mode. Students taught programming in CABLE showed positive attitudes towards the collaborative elements and also towards the online learning elements of CABLE.

The first study by Chan Mow[11]evaluated the effectiveness of CABLE as a teaching environment for programming within a university context, but did not investigate specifically the types of errors students make in programming. The second study [12] aimed to do this. The second study consisted of analyses of computer programs from 3 levels of Computer Science undergraduate programming courses HCS181, HCS281 and HCS286 at the National University of Samoa. The 3 courses were taught using 4 lecture /tutorial hours weekly and were 1 semester in duration. Assessments for these courses were in the form of programming tasks, tests and homework. The programming language used in these courses was Java and the integrated developer environment (IDE) used was JBuilder. For the second study [12], 2 sets of programs were used in the analyses for each course. Programs from 25 HCS181 students, 28 HCS281 students and 15 HCS286 students were used in the analyses. Computer programs from the 3 classes were loaded into JBuilder and from compilation and running of the programs, the errors generated by the compiler were logged and categorized according to the type of errors which emerged.

Findings from this study indicated that i)most of the simple syntax errors were due to carelessness of students; ii) categorization of errors of the present study into syntax, semantic, runtime and logic revealed that syntax errors made up 94.1 %, semantic errors 4.7% and logic errors 1.2%. Hence errors in the 2011 study were predominantly syntax errors (94.1%). This is consistent with findings of similar studies by [14], [15] and [16].

The 2011 study [12] highlighted the common errors students in the 3 courses make while 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. The current study aimed to do this by investigating student perceptions on the various aspects of their programming experience, identify perceived areas of difficulty, and provide more up-to-date data than the 2006 study [11].

3. METHODOLOGY

The study is qualitative in nature. The qualitative approach

was appropriate as the interest was on gaining greater insight and knowledge of students' computer programming experience from an analysis of student perceptions. The assumption is that perceptions of self are based on a socially shared reality and are best thought of as accurate reflections of behavior and experience [13].

The target population is students enrolled in HCS181, HCS281, HCS286 and HCS381 for the academic year 2012. The 4 courses were taught using 4 lecture /tutorial hours weekly and were 1 semester in duration. Assessments for these courses were in the form of a final exam, programming tasks, tests and assignments. The programming language used in these courses is Java and the integrated developer environment (IDE) used is JBuilder.

The instrument used in this study had been adapted from a questionnaire developed by Cheng [4] for a similar study titled: "Teaching and Learning to Program: A qualitative study on sub-degree students in Hong Kong". The survey was adapted to reflect the goals of the proposed study and to reflect the goals of the Java courses at NUS.

At the end of the semester, questionnaires were handed out to students in the 4 programming classes (HCS181, HCS281, HCS286, HCS381). In conducting the survey, consent was sought and participants were assured of the confidentiality of the information provided. A questionnaire was given out to the participants with the intention of a follow up interview if and when necessary. The purpose of the follow-up interview was to ensure accuracy of responses by removing any inconsistencies and any ambiguities in the responses and also to provide any clarification needed by the respondents. The survey was implemented by the members of the research team. 18 students in HCS286 completed the questionnaire.

The questionnaire consisted of 36 items (some of which have multiple questions) of which 15 are multiple choice, 13 used the bipolar anchor method where responses were graduated along a 7-point scale between two clearly opposing anchor points. There were also 12 open ended questions for in-depth and detailed responses. The questionnaire was divided into sections A - H reflecting the major goals of the study:

i) A. Background Information; ii)B. Interests and Experiences ; iii) C. Opinions in learning to program; iv) D. Opinions or perceptions on how Java programming is learnt; v) E. Key motivations in learning to program; vi) F. Kinds of assistance adopted; vii) G. Self-evaluation in learning Java programming and viii) H. Opinions in learning to program in the future. (Please refer to appendices for a copy of the questionnaire).

4. DATA ANALYSES

Data from the multiple choice and questions using the bipolar method were analysed using SPSS. Responses from open ended questions were analysed and categorised into themes and to discern any emerging trends. Data would also be presented graphically using bar graphs and frequency tables.

This paper focuses on findings of the analyses of Questions 4 to 13 and attempts to answer the first part of the research question which is to gauge student opinions on interest and experience, level of mental engagement, motivation, with respect to their programming skills and experience. Factor analysis was carried out to determine if it was possible to create an aggregate or summary variable which would represent all 10 variables. Factor analysis on the 10 items (Questions 4 – 13) using a Principal Components procedure indicated that no single factor resolution was possible (refer Table1). Factor analysis yielded 3 factors and hence it was decided that it was best not to aggregate the variables but to analyse them individually. For analysis, each class was divided into two groups low ability (examination scores between 0 and 60) and high ability (examination scores between 61 and 100). Further dimension was added to the analyses by investigating differences between high and low ability and also gender differences in terms of their responses.

Table 1: Table of Factor analyses for Question 4 – Question 13 using Principal components procedure

Question items (4-13)	Component		
	1	2	3
Boring/stimulating	.764	.075	-.473
Nondemanding/demanding	.547	.312	-.624
not difficult/difficult	.191	.753	.012
Not much/read much	.506	.568	.445
Not enagaed/engaged	.356	-.621	.394
Punishing/rewarding	.766	-.112	.284
not much work/much work	.319	.538	.516
Not understand/understand	.874	-.142	-.090
Not motivated/motivated	.925	-.043	-.006
Not enjoy/enjoy programming	.538	-.672	.069

5. RESULTS

As mentioned earlier, the questions investigated student perceptions on variables such as level of stimulation, "demandingness", difficulty, level of engagement, how

rewarding, the amount of work, their level of understanding of Java, motivation, and enjoyment of programming. The 10 questions used a graduated 7 point scale from1 to 7.

In terms of overall perceptions, an inspection of the means of each question indicated that except for the probe on level of engagement, all other items were above the natural midpoint of 4 indicating positive attitudes to their programming experience (Refer Table II). The high values for the means indicated that students found programming difficult, and also involves a lot of work and readings. However despite this, students found their programming experience rewarding and motivating. However, the fact that the mean for the probe on level of engagement is below the natural midpoint (mean = 3.59) is a cause for concern as it seems to indicate that students did not feel sufficiently engaged in their programming environment and could be a point that the lecturer needs to take note of.

Table 2 : Table of Means of Responses Question 4 – Question 13

Probe	N		Mean	Std. Deviation
	Valid	Missing		
Boring/stimulating	17	0	4.29	2.229
Nondemanding/demanding	17	0	4.71	2.114
not difficult/difficult	17	0	4.65	2.149
Not much reading/read much	17	0	4.94	1.560
Not engaged/engaged	17	0	3.59	2.063
Punishing/rewarding	17	0	4.94	1.819
not much work/muchwork	17	0	5.94	1.676
Not understand/understand	17	0	4.29	1.961
Not motivated/motivated	17	0	5.24	2.166
Not enjoy/ enjoy prog	17	0	5.00	2.151
lab_organised	17	0	4.76	2.078

Analyses of the 10 variables using one way Anova did not reveal any significant differences between attitudes of students of high ability and low ability students for nine of the items. There was significant difference between means of high and low ability students for the item gauging level of difficulty (n = 17,*p* = .013) (refer Table III). Closer inspection indicated that low ability students perceived programming as less difficult compared to high ability students. This may be a cause of concern as it may imply that low ability students are unaware of the actual level of difficulty of programming tasks.

Table 3: Table of differences between Low ability and High ability students for responses for Question 4 – Question 13

		Sum of Squares	df	Mean Square	F	Sig.
Boring/stimulating	Between Groups	6.171	1	6.171	1.350	.266
	Within Groups	59.429	13	4.571		
	Total	65.600	14			
Nondemanding/demanding	Between Groups	5.668	1	5.668	1.175	.298
	Within Groups	62.732	13	4.826		
	Total	68.400	14			
not difficult/difficult	Between Groups	25.725	1	25.725	8.387	.013
	Within Groups	39.875	13	3.067		
	Total	65.600	14			
Not much/read much	Between Groups	1.071	1	1.071	.377	.550
	Within Groups	36.929	13	2.841		
	Total	38.000	14			
Notengaged/engaged	Between Groups	1.376	1	1.376	.278	.607
	Within Groups	64.357	13	4.951		
	Total	65.733	14			
Punishing/rewarding	Between Groups	.686	1	.686	.214	.652

	Within Groups	41.714	13	3.209		
	Total	42.400	14			
not much work/muchwork	Between Groups	2.411	1	2.411	.834	.378
	Within Groups	37.589	13	2.891		
	Total	40.000	14			
Not understand/understand	Between Groups	9.011	1	9.011	2.514	.137
	Within Groups	46.589	13	3.584		
	Total	55.600	14			
Not motivated/motivated	Between Groups	2.743	1	2.743	.701	.418
	Within Groups	50.857	13	3.912		
	Total	53.600	14			
Not enjoy/enjoy prog	Between Groups	.525	1	.525	.122	.732
	Within Groups	55.875	13	4.298		
	Total	56.400	14			
	Within Groups	41.304	13	3.177		

One way ANOVA to determine differences in perceptions on the basis of gender did not reveal any significant differences (refer Table IV).

Table IV: Table of differences in responses based on gender

		N	Mean	Std. Deviation
Boring/stimulating	female	8	4.63	2.326
	male	9	4.00	2.236
	Total	17	4.29	2.229
Nondemanding/demanding	female	8	5.38	2.134
	male	9	4.11	2.028
	Total	17	4.71	2.114
not difficult/difficult	female	8	5.25	2.252
	male	9	4.11	2.028
	Total	17	4.65	2.149
Not much/read much	female	8	5.50	1.690
	male	9	4.44	1.333
	Total	17	4.94	1.560
Not engaged/engaged	female	8	2.88	2.031
	male	9	4.22	1.986
	Total	17	3.59	2.063
Punishing/rewarding	female	8	5.13	1.642
	male	9	4.78	2.048
	Total	17	4.94	1.819
not much work/much work	female	8	5.75	2.121
	male	9	6.11	1.269
	Total	17	5.94	1.676
Not understand/understand	female	8	5.00	1.773
	male	9	3.67	2.000
	Total	17	4.29	1.961
Not motivated/motivated	female	8	6.13	1.458
	male	9	4.44	2.455
	Total	17	5.24	2.166
Not enjoy/enjoy prog	female	8	4.88	2.232
	male	9	5.11	2.205
	Total	17	5.00	2.151
	male	9	4.22	2.224
	Total	17	4.76	2.078

		Sum of Squares	df	Mean Square	F	Sig.
boring	Between Groups	1.654	1	1.654	.319	.581
	Within Groups	77.875	15	5.192		
	Total	79.529	16			
nondemanding	Between Groups	6.766	1	6.766	1.567	.230
	Within Groups	64.764	15	4.318		
	Total	71.529	16			
not difficult	Between Groups	5.493	1	5.493	1.205	.290
	Within Groups	68.389	15	4.559		
	Total	73.882	16			
read much	Between Groups	4.719	1	4.719	2.068	.171
	Within Groups	34.222	15	2.281		
	Total	38.941	16			
engaged	Between Groups	7.687	1	7.687	1.908	.187
	Within Groups	60.431	15	4.029		
	Total	68.118	16			
punishing	Between Groups	.511	1	.511	.146	.708
	Within Groups	52.431	15	3.495		
	Total	52.941	16			
not much work	Between Groups	.552	1	.552	.187	.672
	Within Groups	44.389	15	2.959		
	Total	44.941	16			
understand	Between Groups	7.529	1	7.529	2.092	.169
	Within Groups	54.000	15	3.600		
	Total	61.529	16			
motivated	Between Groups	11.962	1	11.962	2.844	.112
	Within Groups	63.097	15	4.206		
	Total	75.059	16			
enjoy prog	Between Groups	.236	1	.236	.048	.830
	Within Groups	73.764	15	4.918		
	Total	74.000	16			
lab_organised	Between Groups	5.628	1	5.628	1.331	.267
	Within Groups					
	Total	69.059	16			

The preliminary findings discussed in this paper attempt to fulfill the first goal of the study which is to gauge student opinions on interest and experience, level of mental engagement, motivation, with respect to their programming skills and experience. In summary the findings of the probes or items investigating student perceptions on their programming experience indicated the following:

1. On the overall, students showed positive attitudes or perceptions on the overall as indicated by an inspection of the means for each of the 10 items. Inspection of the means indicated that although students felt that programming was difficult, demanding, involved a lot of work and reading, they also found programming enjoyable, stimulating, and motivating. The only point of concern was the level of engagement (mean = 3.59).

2. Except for the probe on level of difficulty there were no significant differences between high and low ability students. High ability students rated programming as involving a lot more work when compared to low ability students.
3. There were no significant gender differences in student perceptions on their programming experience.

These preliminary findings need further elaboration and more in depth explanations will be provided upon the completion of analyses of other sections of the study such as i) background information, ii) interests and experiences, iii) opinions or perceptions on how Java programming is learnt; iv) key motivations in learning to program; v) kinds of assistance adopted; and vi) opinions in learning to program in the future.

In particular responses to the open ended questions should provide more meaningful insights to student perceptions which will in turn help teaching staff provide improved quality of instruction in the area of Java programming. Furthermore, completion of analyses of the rest of the survey should provide information to fulfill the other two goals of the study which are i) to identify perceived areas/topics of difficulty in the programming curriculum and ii) identify perceived deficiencies in learning support and the learning environment.

APPENDIX

Questionnaire
Project Title: NUS Computer programming Student Perceptions of their computer programming experience.

As part of a research project aimed at improving programming courses at NUS we are collecting data on student perceptions of their computer programming experience at NUS. Please complete this questionnaire. Your answers will remain confidential.

Instructions
Please use a ballpoint pen to indicate your views by circling or ticking the appropriate response or by adding your comments and suggestions in the spaces provided.

A.Background Information:

Q1. Gender: A) Male B) Female

Q2. Write down your final mark(in %), if any, in the computer programming courses listed below

a) HCS081 _____
b) HCS181 _____
c) HCS281 _____
d) HCS286_____
e) HCS381_____

B. Interests and Experiences:

Q3. Choose the programming language(s) in which you are able to write simple
programs:
A) HTML B) Javascript C) Java D) Visual Basic
E) Others, please specify_____
C. Opinions in learning to program

For each of the questions below, each statement is to be rated on a scale of 1 to 7. The responses are along a scale between 2 anchor points or extreme points of views. You are to choose which point along the scale best represents your feelings or response by putting a tick (✓) for the response in the appropriate box.

	1	2	3	4	5	6	7	
Q4. found programming boring								stimulating
Q5.found programming as non demanding								demanding
Q6. programming not difficult								very difficult
Q7. Did not read much in programming courses								read a lot
Q8. did not feel engaged								engaged
Q9. found programming punishing								rewarding
Q10. there was not much work involved in programming								Lots of work
Q11. did not understand most of Java concepts introduced in lectures								Understood most of Java concepts introduced in lectures
Q12.did not feel motivated to learn programming								Felt very motivated
Q13 did not enjoy programming								Enjoyed programming

Q14. What are your opinions and/or suggestions about the overall arrangements in

laboratory sessions?

	1	2	3	4	5	6	7	
Poorly organised								Well organised

D. Opinions or perceptions on how Java is learnt.
Q15. How do you perceive the learning of Java? Circle your response.
A) Following – getting through the unit
B) Coding – learning to code
C) Understanding and integrating – learning to write a program through
understanding and integrating concepts
D) Problem solving – learning to do what it takes to solve a problem
E) Participating or enculturation – Discovering what it means to become a
Programmer
F) Others – please specify_____
E.Key motivation in learning to program

Q16. What motivates you to learn to program? Circle those which apply.

A) I want to succeed in the programming class
B) I want to show other students I can program.
C) I want to take up a career in programming
D) I enjoy programming.
E) Others – please specify_____

Q17. How do you learn Java programming in classroom? Circle those which apply:
A) active listening B) pre-study C) more interaction with the teacher
D) understanding individual concepts first E) getting the whole picture first
F) others- please specify_____

F. Kinds of assistance adopted

Q18. What kinds of assistance were you provided in your programming course? Circle your response(s).
a) Learning materials (notes/books) (language?)
b) Programming tools (NotePad, TextPad, JDK, JBuilder, BlueJ, Eclipse, API
Pages)
c) Teachers' and/or peers' assistance (face-to-face, email, phone discussions)
d) Other aids (online tutorials, e-books, program examples, etc.)
e) Others –please specify_____

Q19. i)What kinds of assistance do you need in the learning experiences mentioned in Q18) above?
a) Learning materials (notes/books) (language?)
b) Programming tools (NotePad, TextPad, JDK, JBuilder, BlueJ, Eclipse, API
Pages)
c) Teachers' and/or peers' assistance (face-to-face, email, phone discussions)
d) Other aids (online tutorials, e-books, program examples, etc.)
e) Others –please specify_____

ii) Why are these kinds of assistance needed? List your reasons in the spaces provided below each option.
a) Learning materials (notes/books) (language?)
b) Programming tools (NotePad, TextPad, JDK, JBuilder, BlueJ, Eclipse, API
Pages)
c) Teachers' and/or peers' assistance (face-to-face, email, phone discussions)
d) Other aids (online tutorials, e-books, program examples, etc.)
e) Others –please specify_____

Q20. How long did you spend in this subject per week on average? Circle your response
A) less than 2 hours B) 2 - 4 hours C) 5- 7 hours D) more than 7 hours

Q21. Did you have any learning difficulties or problems? If so, what are they?

Q22. When you found problems during learning to program, what did you do? Circle those that apply.
A) Ask classmates B) Ask teachers C)Study examples in textbooks/Internet
D) Others (please specify)_____

Q23. When you were studying program examples, how did you understand the code?
A) Use program debugger B) Draw flowcharts C) Ask someone D) others – please specify_____

Q24. Which area(s)/topics of the Java language were difficult for you? Circle all those which apply. (You may circle more than one option)
A) Data types and variables
B) Control flow constructs
C) Operators and expressions
D) Classes and objects
E) Inheritance and polymorphism
F) Variables scopes and parameter passing
G) Exception handling
H) Interfaces and abstract classes
I) Graphical user interfaces
J) Threads
K) Packages
L) Data structures, searching and sorting
M) Others - please specify_____

G.Self-evaluation in learning Java programming
Q25. i) How do you see your current ability to program?

A) Poor B) fair C) good D) very good E) excellent

ii) Explain your reason for this assessment of your programming ability?

iii) Provide evidence to support your answer in (ii)?

Q26. i)Can you write a program that works?

A) Yes b) No

ii) How can you check that your program works?

Q27. i) What is a good program?

ii) What procedure do you use to check that your program is a good program?

Q28. What do you think about the readability of your programs? Tick your answer in the space provided.

	1	2	3	4	5	6	7	
Q4 not readable								Very readable

Q29. i) Were there any external factors (eg. workload in other courses, family commitments, sickness, work commitments) that affected your learning in this subject?

A) Workload in other courses
B) family commitments
C) sickness
D) work commitments
E) others – please specify_____

ii) If so, what are these factors and how will they be resolved? Enter your answers in the spaces provided below each option.

A) Workload in other courses

B) family commitments
C) sickness

D) work commitments

E) others – please specify_____

Q30. Did you learn any programming language before learning Java?
i) If so, which language?

A) HTML B) Javascript C) Java D) Visual Basic
E) Others, please specify_____

ii) How do you compare that language with Java?
A) worse B) same C) better D) far better

Q31. How do you compare your performance in Java programming to that of other students in the class?

	1	2	3	4	5	6	7	
Much worse than others								Much better than others

ii) What is the reason for your performance in i) above?

H. Opinions in learning to program in the future
Q32. What should the teacher do to help your learning?

Q33. What should the institution do/provide in helping your learning?

Q34. If you were asked to redesign the programming courses what would you change?

Q35. i)Would the use of the virtual classroom for teaching Java programming be helpful?

ii)What is the reason for your answer in i) above?

Q36. If so, what kinds of functions should be included in the virtual classroom?
A) Recordings of classroom lectures
B) Repository of selected sample programs for illustrating essential concepts
C) IDE(integrated developer environment e.g., JBuilder)
.D) online discussion E) Chat sessions
F)Others? Please specify_____

THANK YOU FOR ANSWERING THE QUESTIONNAIRE.

REFERENCES

1. R.,Naidoo and S.,Ranjeeth. **Errors made by Students in a Computer Programming Course**, Proceedings of the Computer Science and IT Education Conference, 2007.

2. C.,Beard, and J.P.,Wilson. **Experiential Learning**, London Kogan Page, 2005.

3. Slavin, R. **Educational psychology: Theory and Practice**, 7th edn, Allyn & Bacon, Boston, 2003.

4. W.F.J.,Cheng. **Teaching and Learning to Program: A Qualitative study of Hong Kong sub degree students**, Unpublished PHD dissertation, University of Sydney, 2010.

5. J.E.,Ormrod. **Educational Psychology: Developing Learners**, 6th edn, Prentice-Hall, Columbus, 2008, OH.

6. D.H Schunk and Zimmerman of learning and performance man, B. J. (Eds) (1994) Selfregulation. Issues and educational applications. Hillsdale, NJ: Erlbaum, 1994..

7. L.McIver. **Syntactic & Semantic Issues in Introductory Programming education**, Unpublished doctoral dissertation, Monash University, January, 2001.

8. S.M.Thompson. **An Exploratory Study of Novice Programming Errors and Experiences**, Unpublished MSc dissertation. Victoria University 2004.

9. T. Flowers C. A. Carver, and J. Jackson. **Empowering students and building confidence in novice programmers through gauntlet**, in *Frontiers in Education Conference*, vol. 1. ASEE/IEEE, October 2004, pp. T3H/10–T3H/13

10. M. C. Jadud. **A first look at novice compilation behavior using bluej**, in 16th Annual Workshop of the Psychology of Programming, Interest Group (PPIG 2004). Carlow, Ireland: Institute of Technology, April 2004. [Online]. available:http://www.jadud.com/people/mcj/files/2004-PPIG-flcbBlueJ.pdf

11. I.T. Chan Mow. **The Effectiveness of Cognitive Apprenticeship based Learning Environment (CABLE) in Teaching Computer Programming,** Unpublished PHD dissertation, University of South Australia, 2006.

12. I.T. Chan Mow. **Analyses of Student Programming Errors In Java Programming Courses,**. *Journal of Emerging Trends in Computing and Information Sciences*, 3(5) pp. 739-749, 2012.

13. R.R.,McCrae and P.X.Jr Costa, 1989. **Different points of view: Selfreports and ratings in the assessment of personality**. In J. P. Forgas & J. M. Innes (Eds.), *Recent advances in social psychology: An international perspective* (pp. 429-439), 1989. New York: Elsevier Science.

14. J.Jackson, M.Cobb, and C.Carver. 2005. **Identifying top java errors for novice programmers**, Vol. 1, October 2005, pp. T4C–24–T4C–27. [Online]. Available:http://ieeexplore.ieee.org/xpls/abs all.jsp?arnumber=1611967M.

15. M. Ahmadzadeh, D. Elliman, and C. Higgins. **An analysis of patterns of debuggi among novice computer science students**, in ITiCSE 2005:

16. E.,Tabanao, M.M.T.,Rodrigo, and M.,Jadud. **Identifying at-risk novice programmers through the analysis of online protocols,** *Philippine Computing Society Congress, 2008,* (UP Diliman, Quezon City, February 23-24, 2008).