Globus An International Journal of Management & IT
A Refereed Research Journal
Vol 10 / No 2 / Jan-Jun 2019 ISSN: 0975-721X

SELF-EFFICACY IN LEARNING C PROGRAMMING LANGUAGE

*Engr. Ronald John C. Sayson

Abstract

Computer programming is one of the basic skills of students who are in the Information Technology Education field. C Programming Language is the program used in teaching students in their basic computer programming subjects. This study aimed to determine the level of self-efficacy in learning C programming language of students taking up basic computer programming of Carlos Hilado Memorial State College – Alijis Campus. Quantitative type of research was used in this study. Stratified random sampling was employed in the selection of the respondents. The instrument used in assessing C Programming Language Self-Efficacy was adopted from the Computer Programming Self-Efficacy Scale (CPSES) of Ramalingam & Wiedenbeck (1998). The statistical tools used in the study were T-Test for Independent Samples and ANOVA. The level of Self-Efficacy of IT Faculty in their knowledge in C Programming Language is "Very Good". The level of Self-Efficacy in learning C programming language as a whole and when students are grouped according to Age, Sex, and Programs is "Good". There is a significant difference when students are grouped according to programs which reveals that the level of Self-Efficacy of BS in Industrial Technology major in Computer Technology is significantly higher than that of Bachelor of Science in Information Technology.

Keywords: Self-efficacy, Learning C Programming Language, Information Technology Education.

Introduction

A rapid technological advancement in the field of Information and Communication Technology which boost the demands on graduates in the Information Technology Education (ITE), in which the competencies in computer skills are increasing in knowledge based outsourcing around the world. Based on a market analysis shows that the level of investment in software related industries are dramatically increasing compared to hardware related industries, which focus on the learning programming (Lee & Cheng, 2011). As a result, computer programming was integrated in the curriculum in all computer major or related courses in higher education institutions. However, despite and demands on the human workforce in competence and skills in computer programming, there are skill deficiency in computer programming education. In learning computer programming, the process of learning and acquiring the skills are hard and requires a great amount of effort from the teacher. In learning in computer programming, there are level of difficulties experienced and does not understand the content and algorithmic structure of the programming mainly to in-ability to analyze or motivation.

Computer programming is one of the basic skills of students who are in the Information Technology Education field. It is a vital skill that must be learned by every student taking—up computer related subjects. Students will be introduced first to the concepts of computer programming, flowcharting and algorithm. Then start with the first program which prints "HELLO WORLD!" and then the programming skills of students in making programs, analyze the program structure and problem, and give solution to the problem. The student must also analyze and correct or fix the bugs or errors in their program created in order that the program will be properly executed.

C Programming Language is the program used in teaching students in basic computer programming. C programming language is the general purpose language because of its popularity with programmers. C was invented by Dennis Ritchie at Bell Labs in 1972 and in 1978 the publication of The C Programming Language by Kernighan & Ritchie started the revolution in the computing world.

Programs which are taking-up C Programming Language are the Bachelor of Science in

*College of Engineering and Information Technology, Carlos Hilado Memorial State College-Alijis Campus
Bacolod City, Negros Occidental, Philippines

Information Technology (BS Info Tech), Bachelor of Science in Information Systems (BS Info Sys) and Bachelor of Science in Industrial Technology major in Computer Technology (BSIT-CT) of Carlos Hilado Memorial State College – Alijis Campus. The self-efficacy in software programming is very important to be assessed to know if the delivery of the subject is effective or not. Hence, this study was conducted.

Review of Related Literature

A study conducted by Tsai, M. J., Wang, C. Y., & Hsu, P. F. (2019) entitled Developing the Computer Programming Self-Efficacy Scale for Computer Literacy Education aimed to develop an instrument, Computer Programming Self-Efficacy Scale (CPSES), for all students above middle school levels. Based on Berland and Lee's computational thinking framework, this study developed the CPSES items at a literacy level and finally the instrument included the five subscales: Logical Thinking, Algorithm, Debug, Control, and Cooperation. An exploratory factor analysis and reliability tests were conducted in this study. The reliability alpha was .96 for the overall scale, and ranged from .84 to .96 for the subscales. This study also confirmed the positive correlation between computer programming experience and computer programming self-efficacy. In addition, for low- and middle-experienced learners, significant gender differences were found in two subscales: Algorithm and Debug. The CPSES can be applied as an evaluation tool in computer education, robotics education, as well as integrated STEM or STEAM education in which computer programming was regarded as a part of computer literacy.

Yukselturk, E., & Altiok, S. (2017)conducted a study to analyze the effects of programming with Scratch on the views of preservice Information Technology (IT) teachers towards computer programming. The study sample consisted of 151 preservice IT teachers who took an elective course including a Scratch module in the 2013–14 academic year. Three online questionnaires (Personal Information Questionnaire, The Computer Programming Self-Efficacy Scale and The Computer Programming Learning Attitude Scale) were used to gather the quantitative data and focus group interviews were conducted to collect the qualitative data regarding the preservice IT teachers' views in more detail. According to the results, there were significant increases in the mean of the preservice IT teachers' self-efficacy perceptions regarding almost all complex programming tasks after their Scratch programming experience. The results also showed that the preservice IT teachers' negative attitudes towards programming decreased significantly and programming in the Scratch platform had significantly positive effects on the preservice IT teachers' attitudes regarding some items in the scale.

Psycharis, S., & Kallia, M. (2017) conducted a study to investigate whether computer programming has an impact on high school student's reasoning skills, problem solving and self-efficacy in Mathematics. The quasi-experimental design was adopted to implement the study. The sample of the research comprised 66 high school students separated into two groups, the experimental and the control group according to their educational orientation. The research findings indicate that there is a significant difference in the reasoning skills of students that participated in the "programming course" compared to students that did not. Moreover, the self-efficacy indicator of students that participated in the experimental group showed a significant difference from students in the control group. The results however, failed to support the hypothesis that computer programming significantly enhances student's problem solving skills.

Statement of the Problem

The purpose of the study was to determine the level of self-efficacy in learning C Programming Language during the Academic Year 2016 – 2017.

Specifically, it sought to answer the following problems:

1. What is the level of self-efficacy of IT Faculty in their knowledge in C Programming Language?
2. What is the level of self-efficacy in learning C programming language as a whole and when students are grouped according to:
   a. Age
   b. Sex
   c. Programs
3. Is there a significant difference between the level of self-efficacy in learning C programming language when students are group according to:
   a. Age
   b. Sex
   c. Programs

Methods and Materials

Research Design

Descriptive method of research was used in this study.

The following programs and subjects identified in the study are the following:

Programs	**Subjects**
BSIT – CT	COMP TECHNO 6 – Basic Computer Programming
BS INFO TECH	IT PROLOG – Programming Logic and Methods
BS INFO SYS	IS DATAFILE – Data, File and Structure

Respondents of the Study

The purpose of the study is to assess the skills and knowledge of students taking up basic computer programming to selected students of Carlos Hilado Memorial State College – Alijis Campus during the 1st and 2nd Semester, AY 2016-2017.

The participants of this study are the officially enrolled students in their corresponding subjects. 41 Bachelor of Science in Industrial Technology major in Computer Technology (BSIT – Comp Tech) taking-up COMP TECHNO 6 – Basic Computer Programming, 86 Bachelor of Science in Information Technology (BS Info Tech) taking up IT PROLOG – Programming Logic and Methods, and 50 Bachelor of Science in Information Systems (BS Info Sys) taking up IS DATAFILE – Data, File and Structure. A total of 177 students are the respondents of this study. 13 IT Faculty members of CHMSC-Alijis campus were also included.

Research Instrument Used

An instrument on assessing in C Programming Language Self-Efficacy was adopted from the **Computer Programming Self-Efficacy Scale (CPSES)** of Ramalingam & Wiedenbeck(1998) which has 32-items Self-Efficacy Scale for the C++ Programming Language. The research opt to adopt only 27 items.

The following rating scale was used to evaluate each question:

Rating	**Description**
1	Very Poor
2	Poor
3	Good
4	Very Good
5	Excellent

The following mean interval was used:

Scale	**Description**
4.21 – 5.00	Excellent
3.41 – 4.20	Very Good
2.61 – 3.40	Good
1.81 – 2.60	Poor
1.00 – 1.80	Very Poor

Statistical Tool

After gathering and tabulating the data, these were tested and analyzed through a statistical treatment. The statistical software used is SPSS software. Independent Samples T-Test and ANOVA Test were the statistical tools used in the study. Independent Samples T-Test was used in determining the difference of Age (young or old) and Sex (male or female). ANOVA Test was used in determining the difference between the Programs. Tukey was then used to test the significant difference among the Programs.

Sampling Procedure

The study used stratified random sampling with allocation 23% BSIT – Comp Tech, 49% BS Info Tech and 28% BS Info Sys from the 177 students. After computing the Sample Size, the researcher randomly selected 28 BSIT – Comp Tech, 60 BS Info Tech and 35 BS Info Sys students.

Results and Discussions

Table 1 shows the level of Self-Efficacy of IT Faculty in their knowledge in C Programming Language and the overall mean score is 4.15 which has an interpretation of "**Very Good**".

Table 1. Level of Self-efficacy of IT Faculty in their Knowledge in C Programming Language.

No.	Item	Mean	Interpretation
1	I could write syntactically correct C statements	4.23	Excellent
2	I could understand the language structure of C and the usage reserved words.	4.23	Excellent
3	I could write logically correct blocks of code using C program	4.23	Excellent
4	I could write C program that computes the average of any given number of values.	4.38	Excellent
5	I could use built-in functions that are available in the various C	4.23	Excellent

	Libraries.		
6	I could write a small C program given a small problem that is familiar to me.	4.69	Excellent
7	I could write a reasonably sized C program that can solve a problem this is only vaguely familiar to me.	4.15	Very Good
8	I could write a long and complex C program to solve any given problem as long as the specifications are clearly defined	3.85	Very Good
9	I could organize and design my program in a modular manner.	4.08	Very Good
10	I could understand the object-oriented paradigm.	4.38	Excellent
11	I could identify the objects in the problem domain and could declare, define, and use them.	4.15	Very Good
12	I could make use of a pre-written function, given clearly labelled declaration of the function.	4.15	Very Good
13	I could debug (correct all errors) a long and complex program that I had written and make it work.	4.00	Very Good
14	I could comprehend a long, complex multi-file program.	3.69	Very Good
15	I could complete a programming project if I had only the language reference manual for help	4.31	Very Good
16	I could complete a programming project if I could call someone for help if I got stuck.	4.00	Very Good
17	I could complete a programming project once someone else helped me get started.	4.31	Excellent
18	I could complete a programming project if I had a lot of time to complete the program.	4.31	Excellent
19	I could complete a programming project if I had just the built-in help facility for assistance.	4.38	Excellent
20	I could find ways of overcoming the problem if I got stuck at a point while working on a programming project.	4.62	Excellent
21	I could come up with a suitable strategy for a given programming project in a short time.	4.31	Excellent
22	I could manage my time efficiently if I had a pressing deadline on a programming project.	4.08	Very Good
23	I could mentally trace through the execution of a long, complex multi-file program given to me.	4.00	Very Good
24	I could rewrite lengthy and	3.69	Very Good

	confusing portions of code to be more readable and clear		
25	I could find a way to concentrate on my program, even when there were many distractors around.	3.77	Very Good
26	I could find ways of motivating myself to program, even if the problem area was of no interest to me.	3.85	Very Good
27	I could write a program that someone else could comprehend and add features to at a later time.	3.92	Very Good
Overall Mean Score		**4.15**	**Very Good**

Table 2 shows the level of Self-Efficacy in learning C programming language as a whole and when students are grouped according to Age, Sex, and Programs.

Table 2. Level of Self-efficacy in Learning C Programming language as a Whole and when students are grouped according to Age, Gender and Programs

VARIABLE		**MEAN**	**SD**	**INTERPRETATION**
AGE	YOUNG	3.2219	.56579	GOOD
	OLD	3.2851	.51428	GOOD
SEX	MALE	3.3147	.56579	GOOD
	FEMALE	3.1784	.51428	GOOD
PROGRAM	IND TECH	3.4282	.49883	VERY GOOD
	INFO TECH	3.1087	.50775	GOOD
	INFO SYS	3.3394	.58590	GOOD
Overall Score	**Mean**	**3.2471**	**.54297**	**GOOD**

The overall mean score is 3.2471 and its interpretation is "**GOOD**". The Bachelor of Science in Industrial Technology major in Computer Technology got the highest mean score of 3.4282 which has an interpretation of "**VERY GOOD**".

Table 3 to 5 shows the significant difference in the level of Self-Efficacy in learning C Programming Language when students are grouped according to Age, Sex, and Programs.

Table 3. Difference in the Level of Self-efficacy in Learning C Programming language when students are grouped according to Age

Variable	**Mean**	**T**	**Df**	**p**
Young	3.2219	-.631	121	.530
Old	3.2851			

The obtained p-value (p=.530) is not less than .05 level of significance (p>0.05), therefore, accept the null hypothesis. There is no significant difference in the level of self-efficacy in learning C Programming Language and age is not barrier in learning.

Table 4. Difference in the Level of Self-efficacy in Learning C Programming language when students are grouped according to Sex

Variable	**Mean**	**T**	**Df**	**P**
MALE	3.3147	1.398	121	.165
FEMALE	3.1784			

Though the male is higher than female, the difference is not significant.

Table 5. Difference in the Level of Self-efficacy in Learning C Programming language when students are grouped according to Programs

	Sum of Squares	**Df**	**Mean Square**	**F**	**Sig.**
Between Groups	2.367	2	1.183	4.226	.017*

Within Groups	33.601	120	.280
Total	35.967	122	

*Significant at 0.05 level of significance

There is a significant difference when students are grouped according to Programs. Post Hoc Test using Tukey reveals that the level of Self-Efficacy of Bachelor of Science in Industrial Technology is significantly higher than that of Bachelor of Science in Information Technology.

Conclusions

Based on the foregoing findings, the following conclusions are drawn:

1. The level of Self-Efficacy of IT Faculty in their knowledge in C Programming Language is "Very Good".
2. The level of Self-Efficacy in learning C Programming Language as a whole and when students are grouped according to Age, Sex and Programs is "Good".
3. There is no significant difference when it comes to Age and Sex, but there is a significant difference when students are grouped according to Programs.

Recommendations

Based on the conclusions derived from the study, the following recommendations were drawn:

1. Training in C Programming Language to all IT Faculty to better enhance their knowledge and skills.
2. Creation of college module for C Programming Language.
3. The class delivery of 5 hours contact time of BS Info Tech and BS Info Sys must be done at the Computer e more focus on their Programming skills.

References

1. Eranki, K. L., & Moudgalya, K. M. (2016). Comparing the effectiveness of self-learning java workshops with traditional classrooms. Journal of Educational Technology & Society, 19(4),59-74.
2. Mustafa Baser (2013). Attitude, Gender and Achievement in Computer Programming. Middle-East Journal of Scientific Research
3. Özgen Korkmaz (June 2014). Adapting Computer Programming Self-Efficacy Scale and Engineering Students' Self-Efficacy Perceptions. Participatory Educational Research (PER) Volume 1(1), pp. 20-31
4. Petek Askar (June 2009). An Investigation of Factors Related to Self-Efficacy for JAVA Programming among Engineering Students. The Turkish Online Journal of Educational Technology – TOJET Volume 8 Issue 1 Article 3
5. Philip Olu Jegede (2009). Predictors of Java Programming Self-Efficacy among Engineering Students in a Nigerian University. IJCSIS, International Journal of Computer Science and Information Security Volume 4, No. 1 & 2
6. Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. Instructional Science, 45(5), 583-602.
7. Tsai, M. J., Wang, C. Y., & Hsu, P. F. (2019). Developing the computer programming self-efficacy scale for computer literacy education. Journal of Educational Computing Research, 56(8), 1345-1360.
8. Vennila Ramalingam and Susan Wiedenbeck (1998). Development and Validation of Scores on a Computer Programming Self-Efficacy Scale and Group Analyses of Novice Programmer Self-Efficacy. Journal of Educational Computing Research Volume 19, Number 4
9. Yukselturk, E., & Altiok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. British Journal of Educational Technology, 48(3), 789-801.