How to cite this article:


CLOUD-BASED LEARNING SYSTEM FOR IMPROVING STUDENTS’ PROGRAMMING SKILLS AND SELF-EFFICACY

Mohamud Sheikh Ibrahim Abdullahi, Norsaremah Salleh, Azlin Nordin & Ali Amer Alwan
Department of Computer Science
International Islamic University Malaysia, Malaysia
ms.ibrahim@live.iium.edu.my; norsaremah@iium.edu.my; azlinnordin@iium.edu.my; aliamer@iium.edu.my

ABSTRACT

Cloud-based Learning Systems (CBLS) refers to the systems that provide electronic or online content to enable the learning process by offering tools and functionalities through platform available in Cloud. This research seeks to examine the effectiveness of CBLS in improving programming skills among undergraduate students by measuring students’ performance in solving programming problems. This is because there is no empirical evidence on the effectiveness of CBLS when compared with the traditional method of learning programming among student beginners. Traditionally, teaching programming courses has been performed in a classroom setting and it can be very challenging for an instructor to go beyond covering the language’s syntax such as program design skills and problem-solving skills due to the wide variety of students’ background in such bounded class duration. In this study, three single-subject experiments were conducted using 40 undergraduate students enrolled in Web Programming course. The experiments compared the time students spent to solve programming tasks by using traditional learning method and CBLS. A survey to measure students’ self-efficacy was administered before and after the experiments. The
findings of this study showed that there is a statistically significant difference in learning programming using CBLS when compared with traditional method. Our results showed that students solve programming problems in less time when using CBLS. The study also found out that CBLS is effective for improving students’ self-efficacy.

Keywords: Cloud computing, cloud-based learning system, programming skills.

INTRODUCTION

Teaching and learning programming has been a well-known challenging issue faced by students and instructors in higher institutions (Hunter, Livingstone, Neve, & Alsop, 2013; Neve, Hunter, Livingston, & Orwell, 2012). Some students regard computer programming as a difficult subject and less interesting to learn (Hunter et al., 2013; Neve et al., 2012). Similarly, some instructors find it challenging to balance lectures on programming concepts (e.g. conditional statements, loops, functions, arrays and etc.), the language syntax (e.g. PHP, Java syntaxes) and practical aspects to help students develop their programming skills (Hunter et al., 2013). Furthermore, it might become even more difficult for students to learn programming in a large class size and students who are occupied with other commitments (e.g. other courses) (Neve et al., 2012). With the circumstances mentioned above, some studies proposed the need for online and cloud based learning systems that can assist in the teaching and learning programming courses (Hunter et al., 2013; Neve et al., 2012; Wu, Qian, Bhattacharya, Guo, & Hu, 2011).

According to Hunter et al. (2013), a learning system should provide students an ideal environment to learn programming. The learning environment should provide the essential concepts of programming and the relevant syntax to facilitate learning (Hunter et al., 2013). Other than providing concepts and theoretical foundations of programming languages, a learning system typically allow students to write, edit and test their codes (Hunter et al., 2013). Additionally, a learning system is expected to provide relevant features where students are given feedback on the practices they made and their learning progress (e.g. hints and comments on code error) (Hunter et al., 2013).

Conducting an empirical study on possible benefits and impact on learning programming in online and cloud based systems are areas that need more research (Hunter et al., 2013; Neve et al., 2012). There are number of studies that provide evidence on cloud-based learning systems (CBLS) that
have been used in universities for teaching and learning programming (e.g. Selvi & Perumal, 2012; Wu et al., 2011). Some studies suggested that using cloud systems for learning programming had significant impact on students’ learning. For example, studies reported that an online based learning system help improved students’ academic performance in JavaScript and Java programming courses and increase students’ confidence level (Hunter et al., 2013; Neve et al., 2012; Selvi & Perumal, 2012).

Some other studies proposed online based learning systems that can improve students’ self-regulation and self-monitoring (Masud & Huang, 2012; Selvi & Perumal, 2012). Albeit, these previous studies provided descriptive analysis and proposed solutions on how online learning can impact students’ programming skills. In contrast to the online learning or CBLS, teaching programming courses has been traditionally performed in a classroom setting. It can be very challenging for an instructor to go beyond covering the language’s syntax such as program design skills and problem-solving skills due to the wide variety of students’ background in such bounded class duration (Al-Imamy, Alizadeh & Nour, 2006). In this regard, there is a clear lack of evidence on experimental study comparing between CBLS and traditional method of learning programming in terms of effectiveness and self-efficacy. Previous studies did not provide how CBLS can improve students’ programming ability when compared with the traditional method of learning program particularly on learning programming subject among student beginners. Therefore, this research addresses CBLS’ impact on learning programming focusing on two aspects: learning effectiveness and self-efficacy. The contribution of this study is to add empirical evidence on the effectiveness of CBLS in improving student beginners’ programming skills and self-efficacy.

**RELATED WORK**

CBLS is defined as a system that provides “electronic contents to enable the learning process by offering tools and functionalities that assist the learning process on the platform of cloud. All the contents and functionalities of the system should be available as services that can be accessed anywhere through internet” (Selvi & Perumal, 2012). CBLS emerged due to the advancement of cloud computing technology in various domains such as business, manufacturing and education (Kurelović, Rako, & Tomljanović, 2013). For instance, cloud computing provides students and learners access to many applications and services that can be used for formal and informal education (Kurelović et al., 2013). In addition, the use of resources for teaching and
learning is being given flexibility and mobility by cloud computing. Cloud computing also gives greater amount of collaboration and sharing of resources and help creates virtual global communities of teaching and learning (Kurelović et al., 2013).

Advances in technology offer new opportunities in enhancing teaching and learning. The new technologies enable individuals to personalize the environment in which they work or learn, a range of tools to meet their interests and needs. In this paper, we try to explore the salient features of the nature and educational potential of cloud-based application in teaching and learning in a higher education context. In particular, in this study we used Codecademy as one of the cloud-based online learning tools available for students to learn about coding. Codecademy offers several advantages for new learner or programming beginners. Not only that it is free, it simplifies the process to perform necessary settings or configurations prior to start writing codes. Additionally, it offers various programming languages and topics to choose from, and online public forum providing more flexibility for learners to grasp new programming concepts and techniques. According to Shaw (2012), programming language learning with online forum support could help increase students’ learning performance. This is because such an online platform could accommodate different learning styles and different types of participation (Shaw, 2012).

There are a few existing studies looking at the impact of online learning towards programming and problem-solving skills. For example, Pullan, Drew, & Tucker (2013) studied the effectiveness of Java Programming Laboratory (JPL), a cloud-based integrated learning environment for teaching introductory Java programming at Griffith University, Australia. Their findings indicate that JPL approach to teaching introductory programming has been very promising, i.e. impacting positively on students’ learning outcomes. In Ding & Cao (2017), they presented RECT, a cloud-based learning tool that provides learning resources and virtual collaborative computing environment. Their preliminary findings showed that the system can potentially enhance students’ abilities in software project development including debugging experience.

CBLS has a number of benefits such as diversity, flexibility, and accessibility of learning content to the online learners (Masud & Huang, 2012). It has been mentioned by (Selvi & Perumal, 2012) that CBLS systems will become one of the primary mechanisms for teaching and learning in the new era. Therefore, they need to be more effective in terms of interactivity with learners to engage in the learning and teaching processes. It is important to note that existing CBLS systems potentially differ from traditional e-learning systems that have been used today in many institutions (Selvi & Perumal,
Most traditional e-learning systems are known as content centric and they do not provide such a friendly and personalised interface environment for learners particularly student beginners (Selvi & Perumal, 2012).

With the emergence of CBLS as a new tool for learning, learners as well as educators can easily focus on the course content and the education process (Huang, Shu, Chang, Huang, Lee, Huang, & Liu, 2014; Selvi & Panneerselvam, 2012). Further research is required to study how CBLS and online learning systems can be effectively utilized and benefit students (Selvi & Perumal, 2012). According to (Neve et al., 2012), one of the cloud computing areas that need more research consideration is applying cloud computing in a way that students can learn programming better. Learning and teaching programming can become a difficult task for learners and educators (Neve et al., 2012). Study showed that introductory computing and programming classes typically do not plea to many of the beginners (Wu et al., 2011). This is because some students, particularly programming beginners tend to visualize programming concepts as difficult concepts to learn and this may reduce learners’ interest in a deeper understanding of the abstraction of computer science courses (Wu et al., 2011).

**METHODOLOGY**

In this section the methodology of the study is detailed out. The section presents the research objectives, research questions and the hypothesis of the study. It also detailed out the participants, variables, instrumentations and experimental procedure that have been used in this study. Finally, the section concluded with the analysis procedure of the study. The present study aims to achieve two main objectives: i) to investigate the CBLS’ effectiveness for programming beginners; and ii) to survey the effects of CBLS on students’ self-efficacy when solving programming problems.

Based on the research objectives, two research questions were formulated: i) Does CBLS effective in improving students’ programming ability in terms of the time taken to solve the programming tasks? ii) Does CBLS improve students’ self-efficacy in solving programming problems?

To answer the first research question, three experiments were conducted to compare students’ effectiveness in solving programming problem when using CBLS and the traditional method. The time taken by the students in solving PHP programming problems were recorded when using CBLS. Comparison on learning’s effectiveness was made when students are using CBLS and when using traditional method. We conducted a survey at each experiment to measure students’ self-efficacy in addressing the second research question.
Variables

The study has two dependent variables. The first dependent variable is the time (duration) the students spent in solving programming tasks. The study’s independent variable for the first dependent variable contains two related groups: i) students using CBLS, and ii) students using traditional learning method. The second dependent variable is students’ level of self-efficacy in programming. The independent variable’s two related groups for the self-efficacy variable are students’ self-efficacy before using CBLS and after using CBLS.

Hypotheses

To investigate the effectiveness of CBLS in improving students’ programming skill, the following hypotheses were proposed:

Hypotheses 1: Using CBLS will have positive effect on students’ time in solving programming problems compared to traditional method.

This hypothesis is proposed to evaluate whether or not CBLS had positive impact on students’ ability in solving programming problems. It was tested using the time students spent in solving programming tasks while they are using a CBLS against the time they spent on the traditional learning method. In this study, Codecademy was used to represent CBLS whereas the offline editor was used to represent the traditional learning method.

The questions from Codecademy and normal tutorial questions are considerably similar in terms of level of difficulty. The subjects went through three different experiments from three different topics (concepts) of programming. Students were evaluated on three important concepts in programming mainly loops, functions and arrays in both methods (Traditional and CBLS). It is necessary to mention that this research utilised single-subject research design where similar groups are used for control and experimental group. This is sometimes referred to as a repeated measures design (William, 2005). This means that each group become the control and experiment group for themselves. To analyse such experiments where comparison is made regarding the outcomes of both groups, a paired t-test is suitable to evaluate in this situation (William, 2005).

Hypotheses 2: Using CBLS will enhance students’ self-efficacy level in programming.
Students’ level of self-efficacy was evaluated before going through the experiments and after the experiments. This was done to evaluate CBLS’ effectiveness in increasing students’ self-efficacy in learning programming. The study of (Davidsson, Larzon, & Ljunggren, 2009) showed that students’ self-efficacy in programming can contribute to the students’ willingness and motivation in attempting more programming tasks and hence can contribute to the level of programming comprehension. Due to the reasons mentioned above, students’ self-efficacy is important for improving students’ programming skills.

The hypotheses in this study have been investigated in three single-subject experimental designs which were carried out at the Faculty of Information and Communication Technology, International Islamic University Malaysia (IIUM) in Semester 1, 2015/2016 involving an undergraduate course: Web Programming I (INFO 2301). INFO 2301 is an introductory course for first and second year students that consist of learning basics of PHP, MYSQL, HTML and CSS. Prior to conducting the experiment, we obtained permission from the lecturers of the course (Web Programming I) to employ the students in the experiments.

**Research Process**

The planning of this research was made based on the research gaps discovered from the systematic literature review conducted in (Ibrahim, Salleh, & Misra, 2015). Experimental design, the procedures and the instrumentations used in this research were identified at early stages of the study. This followed by the process of gaining approval from the lecturers and tutors of the Web Programming course. After getting the lecturers’ consent, an overview about the experiment was explained to the study participants to make them understand the process of the experiment. The learning and practicing of programming exercises using the cloud based learning systems (Codecademy) took place at the beginning of the experiment. Students were given ample time to practice some exercises to ensure that they were comfortable in using Codecademy. The experiments were conducted starting from week 5 of the semester to week 7 (3 experiments in total). Finally, a report was prepared based on the analysis of results from the experiments.

**Participants**

The participants of our experiment were selected based on convenient sampling process. The subjects of the study are chosen from undergraduate students enrolled in the Web Programming I (INFO 2301) course at ICT faculty in
IIUM. In total, there were 102 students enrolled in three (3) Sections of the course. The experiments were conducted during regular tutorial sessions. Even though there were 48 students who have given their consent to participate in the experiment, only 40 participants successfully attended all the three sessions and participated in the experiments. Therefore, data from these 40 students were used for analyses in this experiment. These participants were undergraduate students who did not have any prior programming experience.

**Instrumentations**

In our experiment, we have utilized the following instruments.

a) **Student’s Consent Form:** The students’ details and consent letter comprised a cover letter and the class lists. The class list is a copy of attendance sheet of students’ basic details such as students’ matric number and year of study. The cover letter is given to provide an overview of the experiments and conditions of the experiment. Students who agreed to participate in the experiment were required to sign the consent form.

b) **Cloud-based Learning System:** Students were asked to use and attempt programming problems in a cloud based learning system. There are a number of cloud and online based learning systems available for learning programming. The Codecademy was chosen as our surrogate measure of CBLS and under this platform, the students used PHP programming. Codecademy was chosen in this study because it does not require subscription fees and also easy to use when compared with few other CBLS which commonly require additional efforts to be familiar with the exercises (Neve et al., 2012). TeamtreeHouse is an example of a CBLS that has a great environment for practicing and learning programming, however it requires monthly subscription fees to use it. While SoloLearn is free CBLS and it has great environment to learn programming, it was not suitable for this study as it strictly requires users to go on sequential basis where users are required to complete each programming topic before moving on to the next topic or exercise. Codecademy was chosen as it is an interactive platform that offers several programming languages for learners to learn for free and allows users to work on any topic and exercise without having to complete or work on any previous exercise or lesson (Neve et al., 2012). Codecademy provides many programming questions available for online learners to practice.
c) Report Forms: A form detailing the time spent in solving problems was used to record students’ progress. This form has been used to record the time spent by the students to solve the programming tasks.

d) Self-efficacy Questionnaire: A questionnaire was used to gather students’ self-efficacy feedback on whether or not CBLS help to improve their self-efficacy level in solving programming problems. We adopted computer programming self-efficacy scales developed by Ramalingam & Wiedenbeck (1998). It was reported that the new scale for programming self-efficacy consisted of 32 items and the reliability of the scores was 0.99. The self-efficacy was measured using a seven-point Likert scale (see Appendix 1 for the self-efficacy survey we used in this study).

Experimental Procedure

This research used a single-subject design as the experiment method (Creswell & Creswell, 2018)]. The three experiments in this study followed the same experimental procedure. At the beginning of the experiment, the participants were given an overview of cloud based learning systems and an explanation on how to use Codecademy (representing CBLS in our study). The experiments were carried out as part of Web Programming I (INFO 2301) tutorial sessions. This course is designed for the first and second year undergraduate IT students, which comprised of fourteen weeks of classes and for each week there were two compulsory classes and a tutorial session.

![Figure 1. Experimental Procedure.](image_url)

All the experiments were conducted during the tutorial sessions run by a tutor. During the tutorial sessions students were asked to use Codecademy and went through some lessons and exercises as part of the tutorial’s activities. They were also asked to solve programming problems using traditional method (i.e.
offline editor) during the tutorial sessions. Figure 1 illustrates the experimental procedure used in this study.

**Analysis Procedure**

This research employed a single-subject research design where the same participants were used for both control and experimental group. This is sometimes referred to as a single-subject research or repeated measures design (Creswell & Creswell, 2018). To analyze such experiments we used statistical tests known as paired t-test and Wilcoxon signed-rank test to compare the performance of both groups (William, 2005). Using these tests, a comparison is made between the actual difference between two means in relation to the variation in the data (measured as the standard deviation of the difference between the means) (William, 2005). The time students spent in solving programming problems using traditional method was compared with the time they spent when solving problems using CBLS. In both sessions, students were given different set of questions of about similar difficulty level. Similarly, students’ self-efficacy level before and after using CBLS was also compared. The comparison and analysis have been done utilizing statistical software, SPSS version 22.

**DATA ANALYSIS AND PRESENTATION OF RESULTS**

This section provides analyses for the experiments and it also provides explanation of the research results. The section starts with providing background information of the participants; followed by the experimental results as well as brief discussions of the experiment results.

**Demographics information of the participants**

In this single-subject experimental design, a total of 40 students from the Faculty of ICT, IIUM took part in the experiment. The participants were first and second year students enrolled in Web Programming I (INFO 2301) course. The design of the experiment was intended to evaluate if CBLS has an effect in improving students’ programming and self-efficacy in solving programming problems. Twenty-four (24) of the subjects were male students while 16 students were female (see Table 1). Although the majority of the participants are Malaysian (57%), there were six other nationalities involved in the experiment including: Afghanistan, Guinea, Nigeria, Syria, Thailand, and Kosovo.
Table 1

<table>
<thead>
<tr>
<th>Gender</th>
<th>Frequency</th>
<th>Percentage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
<td>16</td>
<td>40.0</td>
</tr>
<tr>
<td>Male</td>
<td>24</td>
<td>60.0</td>
</tr>
<tr>
<td>Total</td>
<td>40</td>
<td>100.0</td>
</tr>
</tbody>
</table>

The First Experiment: Loops

The first programming concept in which the participants of the experiment were examined was loop. Students were asked to solve loop related programming problems using their offline editor and then the time taken was recorded. When each student completed the task, results were checked if they matched the correct answer. The tutor verified the answers of traditional method experiment when students completed in solving the problems. Once the student finished the task the time taken to solve the problems was recorded in a timing sheet. After all subjects completed this task, they were asked to work on different programming problem using Codecademy within the same tutorial session. Once students finished the program they can submit the result and the system will notify if the correct answer was submitted. As soon as the answer is verified the time students finished the task was recorded. The time was manually recorded and then saved into Excel sheet. Even though, the questions used in both methods are loop questions, they were different types of loops and the problems being solved were different.

Prior to analyzing the results, the normality of data was checked to ensure the fourth assumption of paired t-test was fulfilled. The data was plotted using boxplots to check outliers in the data to fulfill the third assumption. There were two tests for checking normality namely: Shapiro-Wilk and Kolmogorov-Smirnov (William, 2005). For dataset less than 2000 elements, the Shapiro-Wilk test is used, otherwise, the Kolmogorov-Smirnov test is used (William, 2005). In this case, since there were only 40 elements, the Shapiro-Wilk test is used. To consider a data is normally distributed the ρ-value should be greater than 0.05 (William, 2005). The ρ-value of Shapiro-Wilk is 0.578 when students used traditional method and 0.584 when students used CBLS, therefore our data is normally distributed. Noted that there were no significant outliers in our data.
Output Analysis for the First Experiment (Loops)

The data collected from students during the first experiment were the time spent in solving programming problems using traditional method and CBLS. The analysis showed that students tend to do well in CBLS with 12 minutes on average spent on solving the question while they took 15 minutes to solve programming problems using traditional method. The ρ-value Sig (2-Tailed) tells us whether the means of the two approaches are statistically significantly different (William, 2005). If the ρ-value is greater than 0.05 then it can be concluded that there is no statistically significant difference between the two conditions (William, 2005). That is, the differences are likely due to chance and not likely due to the independent variable manipulation (William, 2005). Our results showed the ρ-value = 0.000, which indicate that there is a statistically significant difference between the two approaches (William, 2005). Table 2 shows the results from paired sample statistics of our data.

Table 2

<table>
<thead>
<tr>
<th>Loops</th>
<th>N</th>
<th>( \chi )</th>
<th>SD</th>
<th>( \chi_d )</th>
<th>SD_d</th>
<th>Paired Test Sig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>40</td>
<td>14.77</td>
<td>3.44</td>
<td>2.77</td>
<td>2.93</td>
<td>0.00</td>
</tr>
<tr>
<td>CBLS</td>
<td>40</td>
<td>12.00</td>
<td>3.35</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Therefore, based on our sample data, we found that there is a statistically significant difference between the mean of programming time spent in traditional method and programming using CBLS. These results showed that students performed better when using CBLS when compared with the traditional method.

The Second Experiment: Function

The second experiment was conducted focusing on functions (methods) related topic. Participants were asked to solve programming problem using traditional method followed by using CBLS. The procedure used in the second experiment was similar to the first experiment, in which to compare the time the students spent when the students solve programming problems using traditional and CBLS method. For the CBLS session, we used different programming questions of the same topic (Functions). Before interpreting the results, the normality of data was checked and the distribution of data was plotted using boxplot to check if data has outliers. The same 40 participants
were involved in this second experiment; hence Shapiro-Wilk is used for analysis (William, 2005). For the traditional and CBLS approach, data is normally distributed with ρ-values (0.104) and (0.360) respectively. Even though, the data in the second experiment contain outliers however, they were not significant outliers that can distort the results of the data.

**Output Analysis for the Second Experiment (Functions)**

In the second experiment, results showed that the students solved functions related programming problem using traditional method in average of approximately 13 minutes (M=12.62). We found that they solved function problems using CBLS with average time of less than 12 minutes (M=11.70) as shown in Table 3. The results from statistical t-test showed that there was a statistically significant difference between the mean minutes of solving problems for the CBLS and the traditional method (i.e. Sig (2-Tailed) value is 0.010, ρ < 0.05). Since the paired samples results revealed that the mean amount of time spent for the non-CBLS method was greater than the mean for the CBLS method, it can be inferred that when the participants were using CBLS method, they were able to solve problems significantly in shorter time when compared with the traditional method. Table 3 shows the results from the paired sample statistics for the second experiment.

Table 3

*Paired Samples Statistics (Functions)*

<table>
<thead>
<tr>
<th>Functions</th>
<th>N</th>
<th>M</th>
<th>SD</th>
<th>M - M</th>
<th>SD</th>
<th>Paired Test Sig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>40</td>
<td>12.62</td>
<td>2.14</td>
<td>0.92</td>
<td>2.15</td>
<td>0.010</td>
</tr>
<tr>
<td>CBLS</td>
<td>40</td>
<td>11.70</td>
<td>2.68</td>
<td>0.92</td>
<td>2.15</td>
<td></td>
</tr>
</tbody>
</table>

**The Third Experiment: Arrays**

The third, which was the final experiment, focused on the arrays topic. In this experiment, participants were given array related programming questions to be solved using traditional method and another set using CBLS Codecademy. Similar to the previous experiments, we recorded the time spent in solving the programming tasks both when using traditional and CBLS. Before the results were interpreted the normality of the data is checked to identify if the data contain outliers. We used Shapiro-Wilk method to test the distribution of data and obtained ρ-value = 0.313 for the traditional method, and ρ-value = 0.324 for the CBLS. These results indicated that the data were normally distributed. There was no significant outlier on our sample data.
Output Analysis for the Third Experiment

As shown in Table 4, the students solved array questions using traditional method approximately in 20 minutes ($\bar{x}=20.27$) while they have spent 18 minutes ($\bar{x}=18.27$) when using CBLS. The results from the paired t-test showed that there was a statistically significant difference between the performance of students using traditional method and CBLS ($p$-value = 0.000, $\alpha < 0.05$). Hence, participants were more effective in solving arrays problems when they are using CBLS.

Table 4

Paired Samples Statistics between Normal and CBLS (Arrays)

<table>
<thead>
<tr>
<th>Arrays</th>
<th>N</th>
<th>$\bar{x}$</th>
<th>SD</th>
<th>$\chi_d$</th>
<th>SD$_d$</th>
<th>Paired Test Sig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>40</td>
<td>20.27</td>
<td>2.61</td>
<td>2.11</td>
<td>2.95</td>
<td>0.000</td>
</tr>
<tr>
<td>CBLS</td>
<td>40</td>
<td>18.27</td>
<td>2.84</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CBLS and Students’ Self-efficacy

To answer the second research question, an experiment was conducted to assess students’ self-efficacy before experimenting CBLS and after using it. Participants were given the same set of survey on both occasions (see Appendix 1). They were asked to rate their self-efficacy level before going through CBLS experiments and after performing the three experiments. To measure the self-efficacy variable, we have adopted self-efficacy instruments developed by (Ramalingam & Wiedenbeck, 1998).

The programming self-efficacy questionnaire consists of 32 items and all questions employed a seven-point Likert scale. The participants were asked to rate their self-efficacy on performing PHP programming tasks using a scale of 1 to 7 before the experiments started. Similar questionnaire was used after the experiments where the students rated their self-efficacy based on their experience in using CBLS. Since the data collected was ordinal and not continuous, we compare the means of each of the 32 items based on the subjects’ respondents in both cases before and after conducting the experiment. This means averages scores from the students’ scales before the experiment and after they went through the experiment was considered. To compare students’ self-efficacy scores whether or not CBLS have improved their self-efficacy we have compared the two sets of scores using Wilcoxon Signed-Rank (Larson & Farber, 2014).
4.9 Students’ Self-Efficacy Results Analysis

As can be seen from Table 5 (the descriptive statistics table) the participants’ self-efficacy before the experiment was low with average mean of 2.89. However, after the three sessions of experiments the self-efficacy of the participants has improved with average mean of 4.16.

Table 5

Descriptive Statistics for Students’ Self-efficacy

<table>
<thead>
<tr>
<th>Self-Efficacy</th>
<th>N</th>
<th>Mean</th>
<th>SD</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before_CBLS</td>
<td>40</td>
<td>2.89</td>
<td>.701</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>After_CBLS</td>
<td>40</td>
<td>4.16</td>
<td>.945</td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>

This shows that CBLS has significantly improved students’ self-efficacy. Table 6 provides comparison of participants’ self-efficacy scores before and after the experiments. The results showed that 34 participants had higher self-efficacy scores after going through the experiments. However, 3 participants had lower self-efficacy scores after the experiments and another 3 participants showed no changes in their self-efficacy scores.

Table 6

Wilcoxon Signed Ranks Test

<table>
<thead>
<tr>
<th>Self-Efficacy</th>
<th>N</th>
<th>χ Rank</th>
<th>SD</th>
<th>Sum of Ranks</th>
</tr>
</thead>
<tbody>
<tr>
<td>After_CBLS</td>
<td>Negative Ranks</td>
<td>3a</td>
<td>7.170</td>
<td>21.50</td>
</tr>
<tr>
<td>Before_CBLS</td>
<td>Positive Ranks</td>
<td>34b</td>
<td>20.04</td>
<td>681.50</td>
</tr>
<tr>
<td>Ties</td>
<td></td>
<td>3c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>40</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a. SE_After_CBLS < SE_Before_CBLS
b. SE_After_CBLS > SE_Before_CBLS

Finally, the results from the Wilcoxon Test Statistics (Table 7) showed that the two sets of scores differed significantly with: Z= -4.979, ρ = 0.00. Therefore, we can conclude that there was a statistically significant difference in students’ self-efficacy scores before and after they used CBLS.
DISCUSSION

CBLS systems are designed to provide students with a friendly interface where students can write codes in a more flexible online environment (Selvi & Perumal, 2012). Three experiments were conducted in this research to assess CBLS’s effect on students’ programming ability against traditional programming method. The findings of this study suggest that CBLS tool (in our case Codecademy) can enhance students’ programming development. The experiments were based on three important programming concepts namely loops, functions and arrays.

It was found in this study that CBLS can help students to solve programming problems in shorter time than traditional method counterpart. Our results also showed that CBLS help improve students’ self-efficacy in solving programming problems. According to (Selvi & Panneerselvam, 2012), such an achievement (i.e. students taking lesser time to solve programming problem) can be considered an encouragement for students to learn more on programming. The findings from this research appeared consistent with some of the findings from the previous studies in the literature, particularly those in the area of CBLS and self-efficacy (e.g. (Huang et al., 2014; Neve et al., 2012)). In particular, the study echoes the observations by (Neve et al., 2012).

Neve et al. (2012) reported that bringing CBLS into learning programming courses enhance students’ enthusiasm, experience and confidence. Our findings also corroborate with the findings of (Hunter et al., 2013) where the online learning system in their university has made the students to be more encouraged and motivated to do programming where 90% of the participants felt happy with their learning system (Huang et al., 2014). Even though, our study and the studies of (e.g. (Huang et al., 2014; Hunter et al., 2013)) have considered CBLS has positive impacts on students’ learning programming, we have utilized different methods of research and different variables. Their studies utilized a questionnaire and they have asked students on their feeling of their systems but in our study we have conducted an experiment to compare students’ traditional method of learning programming and CBLS method.

Table 7

Test Statistics

<table>
<thead>
<tr>
<th>SE_After_CBLS</th>
<th>SE_Before_CBLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>-4.979</td>
</tr>
<tr>
<td>Asymp. Sig. (2-tailed)</td>
<td>.000</td>
</tr>
</tbody>
</table>
Another finding of this study is that CBLS helps improve students’ self-efficacy in programming. Self-efficacy is an important and useful attribute for students as individuals’ performance could be influenced by their self-efficacy (Schunk & Pajares, 2002). Students with higher self-efficacy could attempt to more challenging tasks and spend considerably more time and effort in solving problems and they are also more likely to overcome unexpected difficulties while solving problems compared with students with low self-efficacy (Davidsson et al., 2009; Owolabi & Adegoke, 2014; Schunk & Pajares, 2002). In addition, some studies suggest that self-efficacy stimulates academic motivation and learning achievements (Davenport, 2009; Schunk & Pajares, 2002). Students with low self-efficacy are inclined to view tasks as being too difficult than they really are and thus may experience stress and adopt such a narrow vision of a problem and perceived that they are unable to solve it (Davenport, 2009). Students with low self-efficacy are thus less likely to persevere or to seek out new opportunities for learning (Owolabi & Adegoke, 2014; Schunk & Pajares, 2002). Student’s self-efficacy in programming is therefore very important as it can influence students’ learning and it will encourage students to attempt on challenging programming tasks (Davenport, 2009). Results from our experiments indicate that students’ self-efficacy improved after learning using CBLS tool.

We acknowledged that single-subject research has limitations in terms of its external validity. To ensure validity, the experiments were designed to ensure students were given different set of questions when solving the problems using offline session, and then CBLS. To minimize bias, we also performed random assignment of participants to control and experimental groups and that each experiment involved three different topics (loop, functions and arrays). These require the students to acquire the skills and knowledge needed to solve different programming problems.

This study is subjected to several other limitations. One of the limitations of this research is that the results were confounded to the use of Codecademy as sole representative of CBLS. For instance, the findings would be more reliable if other available CBLS tools such as SoloLearn, TeamTreehouse etc. being utilised in the experiment in addition to Codecademy. Due to the limitation in getting participants, we were not able to use different group of students as the control groups. We faced difficulties in recruiting participants from other Universities because of limited access to potentially eligible participants. Similarly, the study did not capture data from other programming languages such as Java, JavaScript and other programming languages commonly used in the industry and/or academia. Using other programming languages in the experiments could help improve the probability of this study to be generalised in a wider context. There were only forty (40) students took part in full
experiments. More sample data is needed to generalise the study in larger population. The final limitation of this study is that our experiments involved students enrolled in the programming course in our University, hence we did not gather data from other institutions. The views and responses from students from other universities might have depicted a broader picture and would thus produce a more meaningful outcome from the results of the study. Hence, future studies should incorporate and conduct the experiments from students of other universities where feasible. Whilst a focused research is advantageous in certain aspects (Creswell, 2003), a broader analysis involving data from several universities would have also provided more generalisable results therein.

CONCLUSION

In this study, we conducted experiments to evaluate the effect of CBLS using Codecademy as surrogate measures to gauge students programming skills. Students’ effectiveness in programming was measured by the time spent in solving programming problems and students’ self-efficacy was compared before and after the experiments measured by self-efficacy survey. Overall, three experiments (using single-subject design) involving 40 undergraduate students were conducted focusing on three programming concepts: Loops, Functions and Arrays. The results showed that students spent less time in all the three experiments when using CBLS. This indicates that using CBLS can significantly save students’ time in solving programming problems. Students’ self-efficacy is also evaluated before and after undergoing the experiments where the same set of questionnaires were utilized in both surveys.

The results of the study indicate that based on our sample data, students’ self-efficacy showed significant improvement after engaged in online learning using CBLS. Using CBLS for learning programming can significantly enhance the learning experience for students in programming courses and this is evident from the sample data analyzed from this research. Such CBLS systems can potentially deliver flexible, self-paced education to the learners (Selvi & Panneerselvam, 2012; Selvi & Perumal, 2012). The CBLS researches and systems’ potential lies not only in improving delivery of programming courses, or CBLS’ effect in students programming or their self-efficacy, but attempting to create a closer corresponding of the learning loop that exists between a human teacher and students. Eventually, one might imagine a personalized learning or an adaptive environment for learning programming that is capable for learning and adapting to each student, giving advice to them and providing meaningful feedback. Utilizing and making such CBLS
systems will not only improve students’ programming experience and learning but it will also encourage students to write program code on their own since all the help they might need are available from the system. Overall, this study contributes in adding empirical evidence on the effectiveness of CBLS in improving student beginners’ programming skills and self-efficacy.

ACKNOWLEDGMENT

We would like to thank the students who have participated in the experiments and the course instructors for allowing us to proceed with the experiment in the tutorial session.

REFERENCES


Appendix 1 (Self-efficacy questionnaire, adapted from (Ramalingam & Wiedenbeck, 1998))

Part 1: Demographic Information

- Date of Birth (dd/mm/yyyy): ____ / ____ / ______
- Gender:
- Nationality:
- On a scale from 1 – 7, how do you rate your programming competency level:

Part 2: Self-Efficacy

Rate your self-efficacy in doing the following computer programming related tasks using a scale of 1 (not confident at all) to 7 (absolutely confident). If a specific term or task is totally unfamiliar to you, please mark 1

<table>
<thead>
<tr>
<th>Not confident at all</th>
<th>Mostly not confident</th>
<th>Slightly confident</th>
<th>Neutral</th>
<th>Fairly confident</th>
<th>Mostly confident</th>
<th>Absolutely confident</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

1. I could write syntactically correct PHP programming statements.
2. I could understand the language structure of a PHP programming language and the usage of the reserved words.
3. I could write logically correct blocks of code using a PHP programming language.
4. I could write a PHP program that displays a greeting message.
5. I could write a PHP program that computes the average of three numbers.
6. I could write a PHP program that computes the average of any given number of numbers.
7. I could use built-in functions that are available in the various computer applications.
8. I could build my own computer applications.
9. I could write a small program given a small problem that is familiar to me.
10. I could write a reasonably sized program that can solve a problem this is only vaguely familiar to me.
11. I could write a long and complex program to solve any given problem as long as the specifications are clearly defined.
12. I could organize and design my program in a modular manner.
13. I could understand the object-oriented paradigm.
14. I could identify the objects in the problem domain and could declare, define, and use them.
15. I could make use of a pre-written function, given a clearly labelled declaration of the function.
16. I could make use of a class that is already defined, given a clearly labelled declaration of the class.
17. I could debug (correct all the errors) a long and complex program that I had written and make it work.
18. I could comprehend a long, complex multi-file program.
19. I could complete a programming project if someone showed me how to solve the problem first.
20. I could complete a PHP programming project if I had only the language reference manual for help.
21. I could complete a programming project if I could call someone for help if I got stuck.
22. I could complete a programming project once someone else helped me get started.
23. I could complete a programming project if I had a lot of time to complete the program.
24. I could complete a programming project if I had just the built-in help facility for assistance.
25. While working on a programming project, if I got stuck at a point I could find ways of overcoming the problem.
26. I could come up with a suitable strategy for a given programming project in a short time.
27. I could manage my time efficiently if I had a pressing deadline on a programming project.
28. I could mentally trace through the execution of a long, complex multi-file program given to me.
29. I could rewrite lengthy and confusing portions of code to be more readable and clear.
30. I could find a way to concentrate on my program, even when there were many distractors around me.
31. I could find ways of motivating myself to program, even if the problem area was of no interest to me.
32. I could write a program that someone else could comprehend and add features to at a later date.