

The Effects of Blended Learning Approach on Student Performance in The First Computer Programming Course: A Case Study at UiTM

Zulaile Mabni^{1*}, Noratikah Shamsudin¹, Sharifah Aliman¹,
Rosmah Abdul Latif²

¹Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia
{zulaile, atikah, sharifahali}@tmsk.uitm.edu.my

²Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA,
Seremban Campus, 70300 Seremban, Negeri Sembilan, Malaysia
rose@tmsk.uitm.edu.my

*Corresponding Author

Received Date: 15 October 2017
Accepted Date: 29 December 2017

ABSTRACT

Learning programming for novice students is hard because students need to acquire a skill through an active construction process in programming. Previous researches done on teaching and learning programming have identified that programming requires problem solving and analytical thinking skills. Many approaches have been proposed in the literature to tackle the learning problems in computer programming. One of the approaches that have become more pervasive in higher education is blended learning. Thus, the purpose of this study is to investigate the effects of blended learning approach on students' performance in the first computer programming course at Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM). The blended learning approach applied for this study is by combining face-to-face learning and online learning using the Learning Management System (LMS) of UiTM named i-Learn. In this study, a cross-sectional population study with stratified sampling was conducted where a sample of 56 selected undergraduate students enrolled for the first programming course were involved. Prior programming background of these students during their diploma or matriculation program are taken into consideration in this study. Two-way ANOVA analysis revealed that,

the interaction between learning approach and programming background are significant. The mean score for students who have prior programming background and were taught using blended learning approach is much higher as compared to the mean score of students who were taught using the traditional face-to-face learning approach. On the other hand, the mean score for the students who do not have prior programming background and were taught using the traditional face-to-face learning approach, is much higher as compared to the mean score of students who were taught using blended learning approach. Thus, the finding of this study indicated that blended learning approach gave positive impact to students who have prior programming background towards their performance in their first computer programming course.

Keywords: *Blended Learning, Programming, Online Learning, Face-to-Face Learning, Learning Management System*

INTRODUCTION

Learning programming for beginners is difficult because students need to obtain a skill through an active construction process (Bergin & Reilly, 2005). The programming skills that are required for the students are problem solving and analytical thinking skills (Ismail et. al., 2010). The challenges of teaching programming are widely recognized since the failure and dropout rate is high in the introductory programming course (Mendes et al., 2012; Bati et al., 2014). One of the pervasive approach to tackle the learning problem in programming is blended learning approach. Thus, the purpose of this study is to investigate the effects of blended learning approach on student performance in the first computer programming course at Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM).

The blended learning approach applied for this study is by combining the traditional face-to-face learning and online learning using the Learning Management System (LMS) of UiTM named *i-Learn*. *i-Learn* is a learning management system for e-learning in UiTM which acts as a platform that supports the teaching and learning process in UiTM. The *i-Learn* system consists of many applications such as student records, user roles, courses application, tests and quizzes application, internal communication, and evaluation system.

To evaluate the effects of blended learning approach, we applied two different teaching approaches which are blended learning and traditional face-to-face learning to teach the first semester C++ programming course for the undergraduate students. Two groups of students were involved where one group was taught using blended learning approach and the other group was taught using the traditional face-to-face learning approach. For the face-to-face learning, students were taught using the traditional way of lectures, reading and practical session. Whereas, for the blended learning approach, students were taught using the combination of e-learning and face-to-face learning. The e-learning includes online activities such as online lecture notes, programming examples, programming solution, past exams, quizzes, evaluation, discussion and group collaboration. Prior programming background of these students during their diploma or matriculation program are also taken into consideration for this study. In addition to that, students' performance is evaluated based on their final examination scores. We compared the performance of the students who were involved in the blended learning approach to the students who received the traditional face-to-face learning approach.

The finding of this study indicated that blended learning approach gave positive impact to students who have prior programming background towards their performance in their first computer programming course.

RELATED WORKS

Many approaches have been proposed in the literature to deal with the learning problems in computer programming. The most common approach used is called "bricolage" (Pollack & Schertz, 2003). Using this approach,

students skip the analysis and design phases and implement computer programs directly on the computer. They develop their programs gradually by testing them on various examples of input. However, with this practice, these students are not able to explain their algorithms (Ben-David Kolikant & Pollack, 2004). As a result, they have difficulties to construct the algorithms and come up with mental or cognitive obstacles and misconceptions about computer programming.

Another approach is constructivist learning theory where, a learning theory is applied to computer programming (Hadjerrouit, 1999; Mead et al., 2006; Wulf, 2005). By using this theory, novice students need to construct a valid model of a computer to deal with the difficulties of learning programming. Moreover, students must acquire higher-order thinking skills, such as analysis, design, analogical thinking, reuse, evaluation, and reflection to be proficient in computer programming. However, only few educators have applied the constructivist theory to the learning of computer programming (Berglund et al., 2006).

Another solution is to use information technologies in the learning of programming. Using the information technologies, appropriate feedbacks on programming assignments can be given to students through the online programming systems, Web-based programming tutors, online learning systems, or any similar applications. However, there are few examples of online learning applications and Web technologies developed for computer programming. Thus, it is not possible to draw general conclusions about the effect of online learning systems and similar applications on computer programming (Conolly & Stansfield, 2007; Hadjerrouit, 2005; Schwieren et al., 2006). In addition, most applications focus on the aspects of technology rather than pedagogical aspects.

Finally, an attractive solution is using the blended learning models, which blends the traditional face-to-face learning and Web-based systems or any similar software (Bonk & Graham, 2006; Hadjerrouit, 2008). A related study on blended learning style was conducted to compare the advantages in terms of students' participation between blended learning style and pure virtual e-learning style (Dodero et al., 2003). The study indicates that information technologies can act as an incentive to improve students' participation during traditional classroom teaching. However, in

a completely virtual classroom, these technologies do not help to increase their participation.

According to Deperlioglu and Kose (2013), combining face-to-face and online learning can enhance learning and optimize seat time. They conducted a study on using blended e-learning model on a course “Data Structures and Algorithms” given at the AfyonKocatepe University, Turkey. The findings showed that the realized blended learning model provided more effective and efficient educational experience rather than traditional face-to-face learning. Even though many researches have been done to solve the learning and teaching problem in computer programming, the problems and difficulties related to the learning of introductory computer programming remain to be researched.

RESEARCH METHODOLOGY

A case study approach is used to investigate the effect on applying the blended learning approach on the students’ performance in the first programming course. The study conducted was a cross-sectional study which was done in the Faculty of Computer and Mathematical Sciences, UiTM.

Target Sample

The target population was the undergraduate students who enrolled for the first C++ programming course named *Fundamentals of Computer Problem Solving* in semester September 2016 until January 2017. By using stratified sampling, the target sample comprised of two groups of undergraduate students with a total of 56 students. 35 students were taught using the blended learning approach and 21 students were taught using the traditional face-to-face learning approach.

Data Collection

Data was collected from 56 students through email questionnaire using Google form. The questionnaires were distributed to all 56 students. The questionnaires comprised of demographic factor such as student’s ID, program code, type of learning and programming background. In addition to the survey questionnaires, the final exam grades achieved by the students were collected to identify their performance of the subject taught.

Method Analysis

The data was analyzed using R program due to imbalance of datasets. Descriptive statistics was used to summarize all data, while two-factor ANOVA is used to test on the interaction between type of learning and prior programming background on the students' final examination score.

RESULT AND ANALYSIS

The analysis of the results indicates the correlation between students' performance and type of learning approach as well as prior programming background.

Descriptive Statistics

Programming Background and Type of Learning

Figure 1 shows that 15 students and 6 students do not have prior programming background in traditional face-to-face learning and blended learning respectively. While the remaining of the students have prior programming background during their diploma or matriculation program.

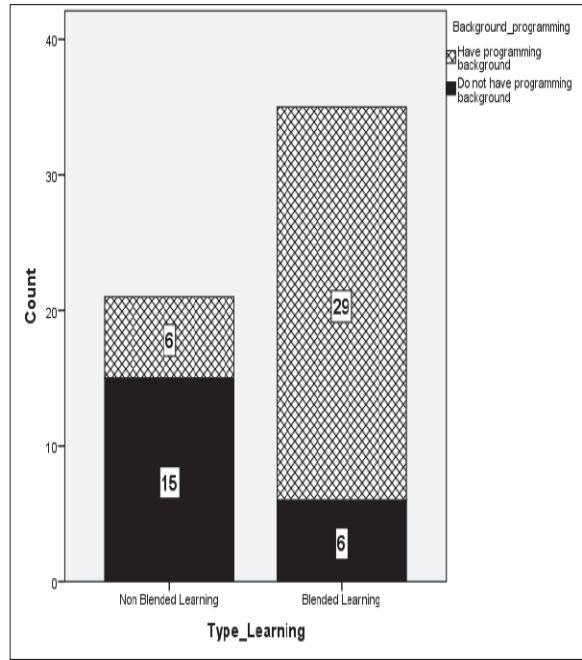


Figure 1: Stacked-bar Chart on Programming Background and Type of Learning

Table 1: Descriptive Statistics of the Final Examination Score

Variables		Background of programming		Types of learning	
		Yes	No	Blended	Non-Blended
Exam Score	Mean	75.800	70.524	75.886	70.381
	Skewness	- 0.010	- 0.520	-0.252	- 0.237
	Kurtosis	- 0.479	- 0.732	0.074	-1.404

From Table 1, in general, we could see that the mean test score for blended learning and have prior programming background are higher as compared to the opposite. The value of skewness and kurtosis is used to indicate that the data is normally distributed according to each level of programming background and type of learning.

Final Examination Scores and Type of Learning

Based on the cluster bar chart in Figure 2 illustrated below, graphically we could see that for blended learning, most of the students with prior knowledge of programming have earned good grades. This contrasted with non-blended learning where, the students without any prior knowledge performed better in the subject taught compared to those with prior programming background.

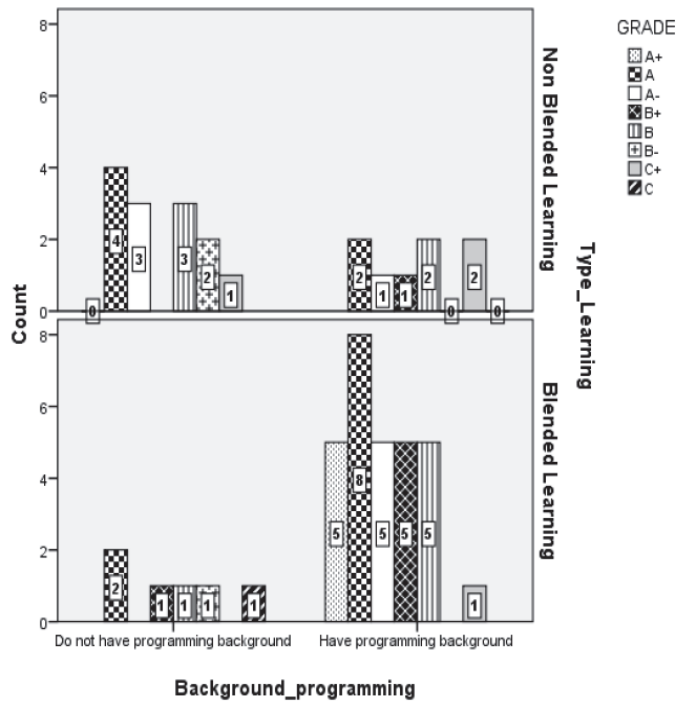


Figure 2: Bar chart on the Students' Final Examination Scores and Type of Learning

Two-way ANOVA

The assumption testing for normality of data, normality of error term and homoscedasticity was done using Shapiro-Wilk test, Normal P-P plot and Levene's test and both assumptions satisfied.

Table 2: Two-way ANOVA for Programming Background and Type of Learning

Variables	Source of variation	Sum squares	df	Mean square	Stats	Sig.
Final test score	Type of learning	397.719	1	397.719	4.265	0.044
	Background of programming	97.407	1	97.407	1.045	0.312
	Interaction	395.946	1	395.946	4.246	0.044
	Error	4849.141	52	93.253		
	Total	5740.213	55			

Table 2 above indicates that the p-value for interaction is significant. This indicates that the interaction between both variables give an impact towards the students' final examination score. As for the main effect, we could see that only the type of learning itself have significant contribution towards the student's final examination score.

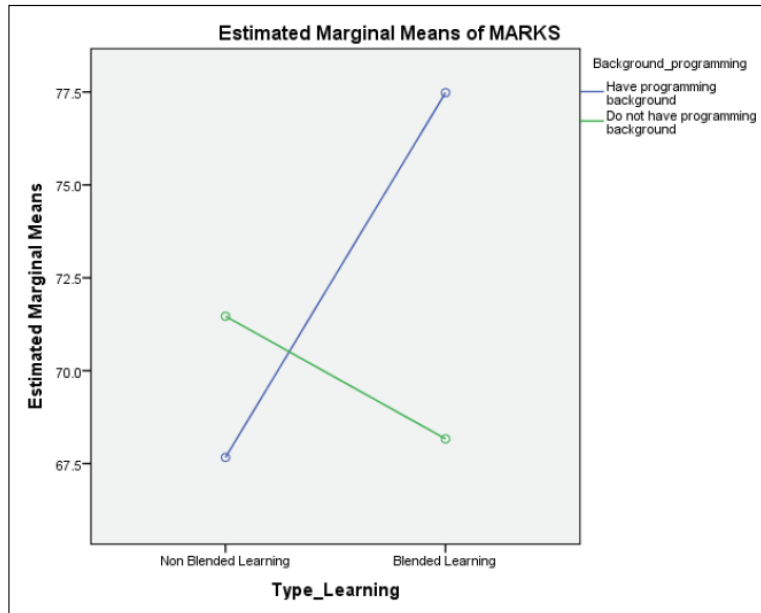


Figure 3: Profile Plot for Interaction between Type of Learning and Programming Background

Furthermore, Figure 3 shows that the mean score for students who have prior programming background and were taught using blended learning approach is much higher as compared to the mean score of students who were taught using the traditional face-to-face learning approach. On the other hand, the mean score for the students who do not have prior programming background and were taught using the traditional face-to-face learning approach, is much higher as compared to the mean score of students who were taught using blended learning approach.

CONCLUSION

As a conclusion, blended learning approach give positive impact towards the students' final examination score including prior programming background knowledge for this case study. In addition, as for traditional face-to-face learning approach, the mean score for students who do not have

programming background is much higher may be due to the convenient of the teaching style as students can approach their lecturers for any uncertainty in the subject taught. For further analysis, the sample size needs to be increased so that predictive analysis can be made in determining the most significant factor that may contribute towards students' performance in programming course.

ACKNOWLEDGEMENTS

We wish to acknowledge the support of Universiti Teknologi MARA (UiTM) under the ARAS Grant No: 600-IRMI/DANA 5/3/ARAS (0013/2016).

REFERENCES

- Bati, T. B., Gelderblom, H., & Biljon, J. V. (2014). *A Blended Learning Approach for Teaching Computer Programming: Design for Large Classes in Sub-Saharan Africa*. *Computer Science Education*, 24(1), 71–99.
- Ben-David Kolikant, Y., & Pollack, S. (2004). *Establishing Computer Science Norms among High School Students*. *Computer Science Education*, 14(1), 21–35.
- Berglund, A., Daniels, M., & Pears, A. (2006). *Qualitative Research Projects in Computing Education Research: An Overview*. In Proceedings of the Eighth Australasian Computing Education Conference (ACE2006), Hobart, Tasmania, Australia, January 2006.
- Bonk, C. J., & Graham, C.R. (2006). *The Handbook of Blended Learning: Global Perspectives, Local Designs*. Pfeiffer Publishing, San Francisco, CA, USA.
- Conolly, T., & Standsfield, M. (2007). *Developing Constructivist Learning Environments to Enhance e-Learning*. In N.A. Buzzetto-More (Ed.), *Advanced Principles of e-Learning*. Informing Science Press, Santa Rosa, California, USA, 19–38.

- Dodero, J.M., Fernández, C., & Sanz, D. (2003). *An Experience on Students' Participation in Blended Vs. Online Styles of Learning*. SIGCSE Bulletin, 35(4), 39–42.
- Deperlioglu, O., & Kose, U. (2013). *The Effectiveness and Experiences of Blended Learning Approaches to Computer Programming Education*. Computer Applications in Engineering Education, 21(2), 328–342, June 2013.
- Hadjerrouit, S. (1999). *A Constructivist Approach to Object-Oriented Design and Programming*. In Proceedings of ITCSE'99, 6/99, Cracow, Poland, 171–174.
- Hadjerrouit, S. (2005). *Web-Based Educational Software in Computer Science: Technical and Pedagogical Usability*. In Proceedings of ED-MEDIA 2005, Montreal, Canada, June 27–July 2, 1139–1144.
- Hadjerrouit, S. (2008). *Towards a Blended Learning Model for Teaching and Learning Computer Programming: A Case Study*. Informatics in Education, 2008, 7(2), 181–210.
- Ismail, M. N., N., Ngah, A., & Umar, I. N. (2010). *Instructional Strategy in the Teaching of Computer Programming: A Need Assessment Analyses*. TOJET: The Turkish Online Journal of Educational Technology – April 2010, 9(2).
- Mead, J. et al. (2006). *A Cognitive Approach to Identifying Measurable Milestones for Programming Skill Acquisition*. In Proceedings of ITiCSE'06, June 26–28, Bologna, Italy, 182–194.
- Mendes, A. J., Paquete, L., Cardoso, A., & Gomes, A. (2012). *Increasing Student Commitment in Introductory Programming Learning*. In 2012 Frontiers in Education Conference Proceedings, 1–6, October 3–6. Seattle, WA: IEEE.
- Pollack, S., & Schertz, Z. (2003). *Supporting Project Development in CS – The Effect on Intrinsic and Extrinsic Motivation*. In Proceedings of the Eleventh International PEG Conference, St Petersburg, Russia.

Schwieren, J., Vossen, G., & Westerkamp, P. (2006). *Using Software Testing Techniques for Efficient Handling of Programming Exercises in an E-Learning Platform*. *Electronic Journal of e-Learning(EJEL)*, 4(1), 87–94.

Wulf, T. (2005). *Constructivist Approaches for Teaching Computer Programming*. In SIGITE'05, October 20–22, Newark, Jew Jersey, USA, 245–248.

