Pairing-Based Approach to Support Understanding of Object-Oriented Concepts and Programming

Shahida Sulaiman

School of Computing, Faculty of Engineering, Universiti Tekonologi Malaysia, 81310 Johor Bahru, Johor, Malaysia
E-mail: shahidasulaiman@utm.my

Abstract— **Blended learning comprises various learning modes with the support of digital resources. It has been a critical element in 21st-century teaching and learning environment at multiple levels of education, mainly tertiary level. In the software engineering field, pair programming is one of the techniques in Xtreme Programming principles in Agile software development methodology. Although pair programming is well-known among practitioners, studies have shown that pair programming can support computer science or software engineering students at higher learning institutions to understand the concepts in programming. Indeed, pair programming could support active learning among students. Inspired by pair programming, this study proposes that pairing-based pedagogy or "pairgogy" in blended learning could also increase students' confidence and interest in completing theoretical in-class exercises not limited to programming tasks with the support of an e-learning system. The proposed approach was applied to teaching object-oriented concepts using Java programs. The findings reflect that both pair programming and "pairgogy" complement each other as a pairing-based approach in blended learning to support understanding of object-oriented concepts and programming. Students' responses towards the approach applied in a semester were positive. The study also implies that most students preferred to be a driver, the person doing the program rather than a navigator who guides drivers on what to program. In this approach, students were also required to complete the in-class theoretical questions in pairs by tracing given programs and answered via the e-learning. Thus, the pairing-based approach has proven to be beneficial to support students in learning programming.**

Keywords— **blended learning; pair programming; pairing-based pedagogy; object-oriented concepts; Java program.**

I. INTRODUCTION

In tandem with the needs of 21st-century teaching and learning, higher learning institutions promote blended learning among their lecturers and students with the support of various e-learning systems. The most popular e-learning system is Moodle. In computer science or software engineering programs, students need to learn both programming concepts and hands-on. Assessments typically include both theories and practical with the ratio of 30:70 or 40:60, where functional assessment is the key in such programming courses. Teaching programming is daunting as students need more attention. Thus, programming classes are generally in small size or with teaching assistants. However, in most cases, lecturers must manage their own programming classes with some efforts to have programming clinics to assist students with difficulties in learning programming. Eliminating this issue, there is a possibility to optimize blended learning through a pairing-based approach that comprises both pair programming (PP) and pairing method with the support of forum feature in e-learning systems.

PP was introduced in the top ten of Extreme Programming principles [1]. PP is more likely a contribution between two persons to complete programming. According to Hanks [2], PP changes individual activity into a collaborative effort. Besides, Plonka et al. [3] believe that "two heads are better that one" idiom to shows the PP is explicitly encouraged in software development because of the value of the contribution. PP also define as a programming activity where two individuals are sitting next to each other, sharing one keyboard and monitor [4]. Umar and Hui [5] state that the PP is a method where two persons work together to solve a given situation. There are two roles in PP, which is a driver and a navigator [6]. The driver focuses on the coding while a navigator actively observes the work by the driver. The following sub-sections include the related works that cover the literature in computing education, specifically in a programming course, followed by the works on PP and blended learning in tertiary education.

A. Programming

Programming is considered a challenging course at the tertiary level. Many works attempt to support students in mastering their programming skills. Wainer and Xavier [7]

attempt to assist students in an introductory programming course by comparing Python with C language. The study adopted a controlled experiment to study the differences. Besides, Marin et al. [8] propose the use of gamification in programming courses. While Lagus et al. [9] suggest transfer-learning methods to predict programming course outcomes. Bosnić et al. [10] studied the impact of the distributed software development course among software engineering fresh graduates. Finally, Yeomans et al. [11] compared the different perspectives between students and professional programmers on challenging concepts in programming. These works contribute to diverse aspects of supporting computing students when mastering programming skills. However, this study focuses on the PP approach. The following sub-section describes the existing works concerning PP.

B. Pair Programming

PP is an effective way to learn programming rather than individual programming. Plonka et al. [3] state that by using PP to solve the given scenario can allow the partners to share knowledge. While sharing knowledge, both partners can achieve meaningful ways to solve the problem regarding the code. PP also influences student's ability to learn the language even it consists of the different ways of the learning style of students [5]. By using PP, both partners may switch roles while doing programming [12].

In PP, the roles are divided into two [4], [6], [12] parts as follows:

- driver who types the code and handle any input devices and
- navigator who follow-ups the code being typed by the driver and suggests better methods and solutions.

Hanks [2] state that PP can help in reducing the gender gap between male and female students in computing-related majors because male students are more confident in their skills than female students even though the competency between them are the same. Since PP is a contribution between two individuals, the partners can physically meet, which can increase productivity, collaboration, avoid stress and reduce travel time as both partners are sitting side by side with the same keyboard and monitor.

There are some examples of criteria used while experimenting with the PP. Firstly, Hannay et al. [13] and Arisholm et al. [6] believe that duration is the time taken to complete all tasks that had been assessed to perform a set of change tasks. Besides, Hannay et al. [13] also state that quality is some test cases or several correct solutions for the tasks, student grades, functionality, and metrics for code complexity. Next, the example of criteria to be measured is an effort that describes the total effort spent in the pair or the total number of programmer hours taken to develop a correct program [1], [6], [13]. Besides, the work by Salleh et al. [15] proves the positive impacts on students' performance when using PP in learning programming skills. Besides, other example criteria that can be measured in PP is teaching strategies. As stated by Plonka et al. [3], teaching strategies can be divided into two groups, which are teaching strategies to use when a novice is driving, and teaching strategies use when an expert in driving. The next sub-section elaborates on the blended learning at the tertiary level.

C. Blended Learning

Blended learning key elements include various learning modes with the support of digital learning materials. In the case of UTM, it promotes the New Academia Learning Innovation (NALI) model [16] among the lecturers to encourage a new way of learning in line with the National Higher Education Strategic Plan (PSPTN). With the support of different learning modes (pedagogy/andragogy/heutagogy) and learning materials (digital resources), new academia adopts a learning innovation model towards student-centered learning that promotes entrepreneurial academia. It includes 30 to 80 percent of online, blended learning.

There are some works related to blended learning, such as the study by Hoic-Bozic et al. [18] that proposes a recommender system and Web 2.0 tools to enhance blended learning. While the proposed approach by Cabrera et al. [19] attempts to blend communities and team-based learning in a programming course. Besides, the work by Pardo et al. [20] aims to predict students' performance by combining self-regulated learning indicators and engagement with online learning events. Leite et al. [21] propose a blended learning method applied in data communication and computer networks subject.

Ghazal et al. [22] provide a model to promote students' experience and satisfaction in a blended learning environment. Besides, Dorobăț et al. [23] suggest a conceptual model that could assess the success of learning management systems in higher education. The work by Ellis et al. [24] attempts to redefine collaboration for engineering students. Concisely, all studies show a good impact on blended learning as compared to the traditional way of teaching. Hence, the work in this paper attempts to prove that the pairing-based approach in blended learning as elaborated in the following section.

II. MATERIALS AND METHODS

This section provides the details of the proposed approach and the study conducted.

A. Pairing-Based Approach in Blended Learning

This sub-section includes the details of the proposed pairing-based approach in which theoretical concepts are conducted using pairing-based pedagogy or "pairgogy" in blended learning. It is subject to increase students' confidence and interest in completing theoretical in-class exercises not limited to programming tasks using PP with the support of an e-learning system. The components in blended learning involve both brick and mortar (classroom) that include face-to-face and online learning using technology using computers or any related devices virtually [17].

In the proposed pairing-based approach, it includes both face-to-face and virtual (online) in the classroom. The classroom here refers to the computer laboratory as the study refers to the course related to object-oriented concepts and programming where labs are required. Students could access the course materials (lecture slides, lab assessments, assignments), post forums, and submit their programming exercises and given assessments via online learning. During each PP-based exercise, students sit in pairs and one of them (the drivers) required to submit the in-class lab exercises by

uploading their source codes and outputs via e-learning. In some theoretical exercises, students work in pairs to trace the given problems manually and discuss before uploading their answers to the e-learning, for instance, by replying to the lecturers' posts. Different groups will reply to the same post that motivates them to think and work faster to provide their answers. Once all answers are available, the lecturer could view how many pairs have traced the programs correctly without coding. The varieties in answers reflect that students do not tend to copy the answers posted by earlier groups as they believe that they have their correct answers after discussing in pairs.

After completing the manual tracing of source codes in the given exercises, students start to use the editor to write the programs, compile and see the outputs. Students will get excited to check whether their theoretical concepts based on the manual tracing answers are correct or not. Thus, the pairing-based approach or "pairgogy" could complement the benefits of PP in achieving the course learning outcomes related to programming. Fig. 1 shows the proposed architecture of the pairing-based approach in a blended learning environment involving both students and lecturers.
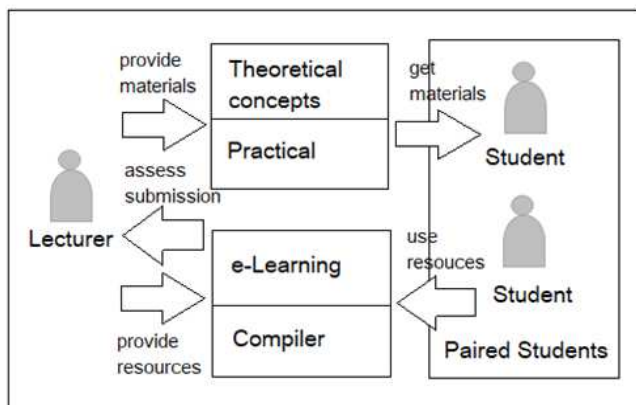

Fig. 1 Proposed architecture of the pairing-based approach in a blended learning environment for a programming course

B. The Study

This sub-section reports the study conducted. In this study, a lecturer provides materials for the theoretical concepts and practical for a particular programming course. The paired students get the materials, mainly the lab exercises, to be done in class. The lecturer also adopts the e-learning to post forums related to paired works on understanding theoretical concepts such as tracing of program outputs manually. Paired students use e-learning to provide the answers to the exercises without using the programming compiler. Then, paired students start to write the programs using the compiler and check their answers. The lecturer should consider exposing to the students the use of an Integrated Development Environment (IDE) to compile the programs as most IDEs proving tracing utility. This will allow students to do the tracing using IDE as compared to their manual tracing done earlier. Then the lecturer can assess students' submissions to see their performance by checking their manual tracing versus computerized tracing using a compiler or an IDE. The longitudinal study comprised second-year Computer Science students during the fourteen-week of a semester period. Twenty-four students involved in the study.

The details of the topics and lab exercises that adopted the PP are listed in Table 1.

TABLE I
DETAILS OF TOPICS AND LAB EXERCISES USING PP

No	Topic	Lab Exercise
1	Introduction to classes and objects	Write a Java program for the given Product class (with attributes and operations)
2	As above	Passing parameter in the method
3	As above	Constructor and method overloading
4	Static members	Static class members
5	As above	Static method
6	Java package enumerated type and wrapper class	Java enumerated type
7	As above	Wrapper class
8	Arrays	Array of objects
9	As above	ArrayList

To introduce object-oriented concepts, the Unified Modelling Language (UML) model that is a class diagram, is also introduced. The first exercise in Table 1 shows an example of how students need to write a Java program based on the given class with attributes and operations. In this case, only PP is applied. The rest of the exercises cover either output tracing or error identification and modification. Due to time limitation and suitability, some exercises only adopted PP, but they are not complemented with manual tracing using "pairgogy". For exercise, no. 9, the exercise is part of the revision for the mid-term exam. In terms of pairing, the random approach was used.

For the first exercise, students could choose any partners they preferred to have and negotiated with their partners in terms of the roles (driver or navigator). For the second exercise, students had to choose the course mates sitting next to them and then swapped their roles for the third exercise. For the fourth and fifth exercises, students could choose their own partners but different from the previous exercise. Students traced manually before coding the exercise using the given IDE. For the sixth and seventh exercises, students had to choose the partners next to them but flipped coins to determine their roles. While for the eight exercises, students had to work with someone sitting next to them and justified why they should be given the role if both partners preferred the same role. For the last exercise, students could choose any partners. For this exercise, students had to also trace the programs manually before coding. All the exercises were from the printed modules. Thus, students could not just copy and paste the programs but they must write the programs together as either a driver or a navigator.

Students were given a certain time to complete their tasks as in programming skill-based tests. Students need to program when given two problems: correcting errors and problem-solving within the given time frame. This also includes the theoretical concepts where students need to trace given programs manually in the given time constraints. As for the exercises, students need to upload their answers via the e-learning, done by the drivers. Some exercises are

considered for formative assessments as part of the overall assessments.

III. RESULTS AND DISCUSSION

The analysis includes the feedback on PP, role preference, and assignment, either as a driver or a navigator. Fig. 2 shows students' responses towards PP in terms of enjoyment, confidence, preference to use PP, and usefulness of PP in understanding difficult topics. It is quite surprising to see one student (5%) did not agree that PP could increase his/her confidence in learning programming. While one student (5%) was not sure concerning the preference to use PP when completing lab exercises. Overall, the study shows PP is a good approach to learning programming in line with earlier studies, as discussed in the related works.
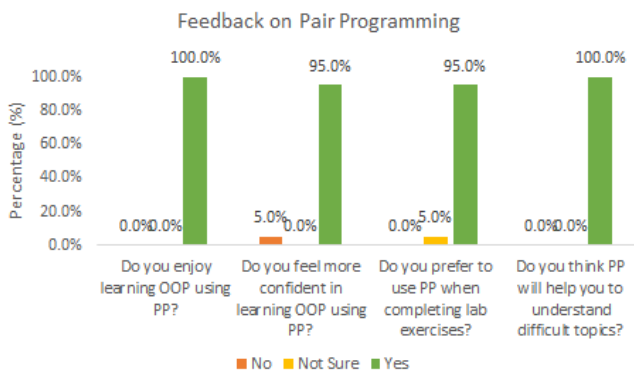

Fig. 2 Students' responses towards PP

Fig. 3 shows students' role preference in terms of being a driver or a navigator. The study shows that most students (64.3%) prefer to be a driver, the one who typed the programs using a computer while following the instructions from the navigator. However, the partners who shared the same preferred roles (stated by one student or 7.1%) had to negotiate to proceed with the given exercises using PP.
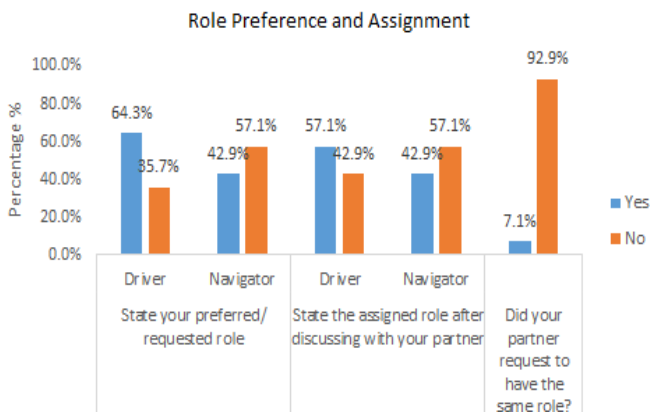

Fig. 3 Students' role preference

This reflects that most students prefer to learn by doing (being a driver), not just instructing (being a navigator). However, the idea of PP is to allow more discussion through pairing works, especially when PP is complemented with "pairgogy" to trace programs manually in each time followed by the coding to check the correctness. The findings reveal that the pairing-based approach in a blended learning environment is good to support 21st-century

learning. This is also in line with constructive learning where students could discuss among them at the same time promoting student-centered learning during face-to-face classes.

As most programming courses nowadays, do not have teaching assistants or tutors, lecturers must assist students in programming themselves. Most universities practiced small classes for programming courses, as helping students to debug their programs is time-consuming. By using the pairing-based approach of PP and "pairgogy", it promotes collaboration among students who discuss, share their knowledge, and correct each other. This approach does not only encourage cohesiveness between the pairs but also among the pairs as they will ask other pairs should they face problems in solving the given exercises. When analyzing the same data by each respondent, they provide a better view of the responses as shown in Table 2.

TABLE II
DETAILS OF RESPONSES ON PREFERRED AND ASSIGNED ROLES

Respondent	State your preferred/ requested role		State the assigned role after discussing with your partner		Did your partner request to have the same role?
	D	V	D	V	
R1	N	Y	N	Y	N
R2	Y	N	Y	N	Y
R3	N	Y	N	Y	N
R4	N	Y	N	Y	N
R5	Y	N	N	Y	N
R6	Y	N	Y	N	N
R7	Y	N	Y	N	N
R8	Y	N	Y	N	N
R9	N	Y	N	Y	N
R10	Y	N	Y	N	N
R11	Y	Y	Y	N	N
R12	N	Y	N	Y	N
R13	Y	N	Y	N	N
R14	Y	N	Y	N	N

Legend: D: Driver, N: Navigator, Y: Yes, N: No

R2 stated that his/her partner requested to have the same role (driver) as R2 preferred the role (stated as Yes). For R5, it shows that he/she preferred the driver role but did not get the role (driver), but at the same time, it did not state that the partner requested the same role. There is a possibility that the partner did not directly state the preferred role but ended not giving the preferred role to the partner. R11 stated that he/she preferred both driver and navigator roles, in other words, being neutral to both roles.

Fig. 4 shows two feedbacks by the students when asking via the forum in the e-learning regarding their partners and their roles, whether PP helped them to understand the programming concepts and how PP managed to help.
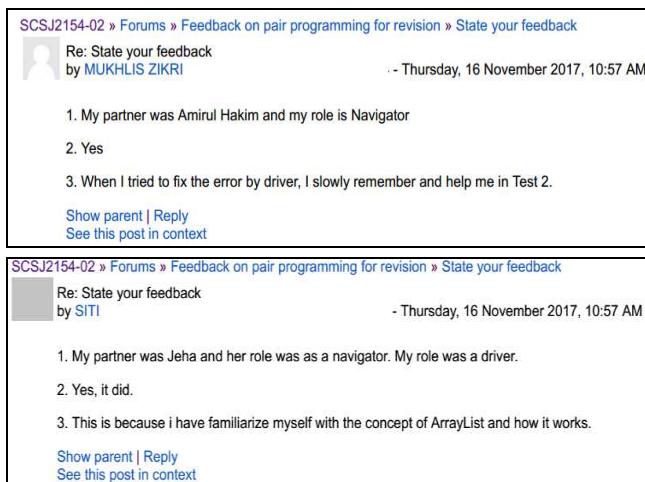
Fig. 4 Students' feedback on how PP could assist their understanding

As compared to PP in the real industry setting, PP in education setting also gives positive impacts mainly in the process of learning. For instance, the first feedback states that when fixing errors, as a navigator he also learned something and could remember the process of learning together. For the second feedback, the role was as a driver. She could understand the concept better when working in pairs. Hence, the early analysis and findings mainly in PP have inspired the use of "pairgogy" in understanding the theory by giving the exercises on tracing programs manually before writing the programs. This has become a good complement of PP in supporting the understanding of object-oriented concepts and programming skills. Fig. 5 shows two different traced outputs given by two different pairs for the same exercise.
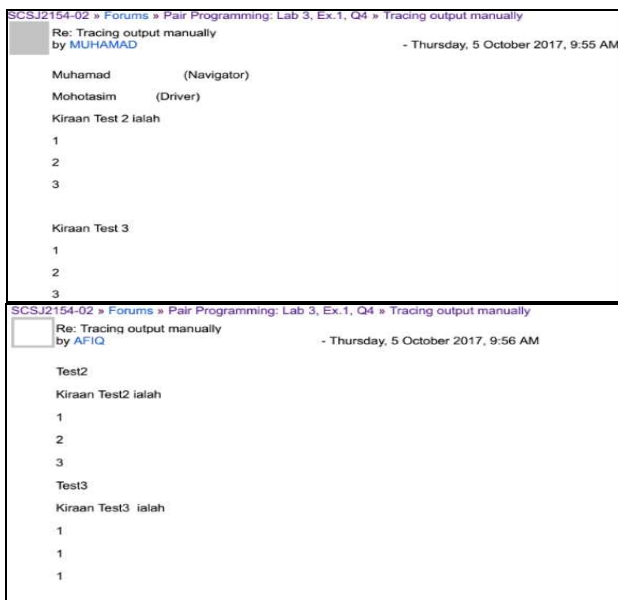

Fig 5 Examples of the manual tracing of outputs answered via a forum

Although all students could see other pairs' answers, they trust their tracing as discussed with their partners. This approach indirectly inculcates the feeling of confidence in students when giving their answers. Once they write the programs, then they can check whether what they have traced manually is correct.

IV. CONCLUSIONS

In short, the proposed pairing-based approach of PP and "pairgogy" can be adapted or adopted in any programming related courses in computer science to support both understanding the theories and practical aspects of programming such as object-oriented. However, it must be with the support of an e-learning environment where lecturers use the forum, for instance, responses on manual tracing before students writing the programs using the compiler and upload their solutions via the e-learning to be accessed. At the same time, lecturers could compare the correctness of the solutions given by the students in terms of manual tracing versus compiled programs.

Future works include the possibility to explore more factors to be concerned in the proposed pairing-based approach that also includes pairing based on genders, good versus average students, and the direct impact in their performance in both theoretical and practical skills in programming.

REFERENCES

[1] W. Sun, G. Marakas, M. A. Urreta, "The Effectiveness of Pair Programming Software Professionals' Perceptions," *IEEE Software*, vol. 33, no. 4, pp. 72-79, 2016.

[2] B. Hanks, "Empirical Evaluation of Distributed Pair Programming," *Int. J. Human-Computer Studies*, vol. 66, no. 7, pp. 530-544, 2008.

[3] L. Plonka, H. Sharp, J. C. D. Linden, Y. Dittrich, "Knowledge Transfer in Pair Programming: An in-Depth Analysis," *Int. J. Human-Computer Studies*, vol. 73, pp. 66-78, 2015.

[4] K. S. Choi, F. P. Deek, I. Im, "Exploring the Underlying Aspects of Pair Programming: The Impact of Personality," *Information and Software Technology*, vol. 50, no. 11, pp. 1114-1126, 2008.

[5] I. N. Umar, T. H. Hui, "Learning Style, Metaphor, and Pair Programming: Do They Influence Performance?," *Procedia - Social and Behavioral Sciences*, vol. 46, pp. 5603-5609, 2012.

[6] A. Arisholm, H. Gallis, T. Dyba, D. I. K. Sjoberg, "Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise," *IEEE Transaction on Software Engineering*, vol. 33, no. 2, pp. 65-86, 2007.

[7] J. Wainer, E. C. A. Xavier, "Controlled Experiment on Python vs C for an Introductory Programming Course: Students' Outcomes," *ACM Transactions on Computing Education (TOCE)* vol. 18, no. 3, pp. 12:1-12:16, 2018.

[8] B. Marín, J. Frez, J. Cruz-Lemus, M. Genero, "An Empirical Investigation on the Benefits of Gamification in Programming Courses," *ACM Transactions on Computing Education (TOCE)*, vol. 19, no. 1, pp. 4:1-4:22, 2018.

[9] J. Lagus, K. Longi, A. Klami, A. Hellas, "Transfer-Learning Methods in Programming Course Outcome Prediction," *ACM Transactions on Computing Education (TOCE) - Special Issue on Learning Analytics and Regular Papers*, vol. 18, no. 4, 2018.

[10] I. Bosnić, I. Čavrak, M. Žagar, "Assessing the Impact of the Distributed Software Development Course on the Careers of Young Software Engineers," *ACM Transactions on Computing Education (TOCE) - Special Issue on Global Software Engineering Education*, vol. 19, no. 2, 2019.

[11] L. Yeomans, S. Zschaler, K. Coate, "Transformative and Troublesome? Students' and Professional Programmers' Perspectives on Difficult Concepts in Programming," *ACM Transactions on Computing Education (TOCE)*, vol. 19, no. 3, 2019.

[12] M. Nawahdah, D. Taji, "Work in Progress: Investigating the Effects of Pair-Programming in Students' Behavior in an Advanced Computer Programming Course," *IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE 2015)*, IEEE, pp. 157-160, 2015.

[13] J. E. Hannay, T. Dyba, E. Arisholm, D. I. K. Sjoberg, "The Effectiveness of Pair Programming: A Meta-Analysis," *Information and Software Technology*, vol. 51, no. 7, pp. 1110-1122, 2009.

[14] J. E. Hannay, E. Arisholm, H. Engvik, D. I. L. Sjoberg, "Effects of Personality on Pair Programming," *IEEE Transactions on Software Engineering*, vol. 36, no. 1, pp. 61-80, 2010.

[15] N. Salleh, A. Nordin, H. S. Yaacob, "Experimenting with Pair Programming in a 2nd Year Java Programming Course," *Proceedings of the Third Malaysian Software Engineering Conference* (MySEC'07), 2007.

[16] NALI (2018). [Online]. Available: http://ctl.utm.my/nali/

[17] TESLA (2018). [Online]. Available: http://www.theteslaacademy.com/whatisblended/

[18] N. Hoic-Bozic, M. H. Dlab, V. Mornar, "Recommender System and Web 2.0 Tools to Enhance a Blended Learning Model," *IEEE Transactions on Education*, vol. 59, no. 1, pp. 39-44, 2016.

[19] I. Cabrera, J. Villalon, J. Chavez, "Blending Communities and Team-Based Learning in a Programming Course," *IEEE Transactions on Education*, vol. 60, no. 4, pp. 288-295, 2017.

[20] A. Pardo, F. Han, R. A. Ellis, "Combining University Student Self-Regulated Learning Indicators and Engagement with Online Learning Events to Predict Academic Performance," *IEEE Transactions on Learning Technologies*, vol. 10, no. 1, pp. 82-92, 2017.

[21] F. N. Leite, E. S. Hoji, H. A. Abdalla, "Blended Learning Method Applied in Data Communication and Computer Networks Subject," *IEEE Latin America Transactions*, vol. 16, no. 1, pp. 163-170, 2018.

[22] S. Ghazal, H. Al-Samarraie, H. Aldowah, "I am Still Learning": Modelling LMS Critical Success Factors for Promoting Students' Experience and Satisfaction in a Blended Learning Environment," *IEEE Access*, vol. 6, pp. 77179-77201, 2018.

[23] I. Dorobăţ, A. M. I. Corbea, M. Muntean, "Integrating Student Trust in a Conceptual Model for Assessing Learning Management System Success in Higher Education: An Empirical Analysis," *IEEE Access*, vol. 7, pp. 69202-69214, 2019.

[24] R. A. Ellis, F. Han, A. Pardo, "When Does Collaboration Lead to Deeper Learning? Renewed Definitions of Collaboration for Engineering Students," *IEEE Transactions on Learning Technologies*, vol. 12, no. 1, pp. 123-132, 2019.