Programming Learning Requirements based on Multi Perspectives

Firas Layth Khaleel[1], Noraidah Sahari @ Ashaari[2], Tengku Siti Meriam Tengku Wook[3], Amirah Ismail[4]

[2,3,4]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, SOFTAM, 43600, Malaysia
[1]College of Computers Science and Mathematics, Computer Science Department, Tikrit University, 00964 - Iraq

Article Info

Article history:

Received Oct 12, 2016
Revised Jan 15, 2017
Accepted Jan 30, 2017

Keyword:

Game elements
Motivation
Multi perespectives
Programming language
Programming requirements

ABSTRACT

Students occasionally find it difficult to learn new programming languages. They often confront unfamiliar programming terms and having difficulty to visualize the processes that happen in computer memory. Weak students find this a burden and end up memorizing the processes without understanding them and their workings. This situation invariably leads students to obtain low grades in their programming subjects. The preliminary investigation of survey was produce in previous work; the results showed that students have experienced ineffective learning, lack of interest towards this course and lack of motivation. Therefore, the main objective of this study is to identify programming learning requirements based on previous work and verified from two perspectives (experts and students). Finally, the result was a group of requirements that should be considered in programming language course.

Copyright © 2017 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Firas Layth Khaleel,
College of Computers Science and Mathematics,
Computer Science Department,
Tikrit University,
Iraq.
Email: Firas_Layth@tu.edu.iq; ukmp69617@siswa.ukm.edu.my

1. INTRODUCTION

Programming Language is an essential undergraduate course for Computer Science, Information Technology, and Software Engineering departments [1]. Programming is not an easy subject to learn because it requires deep and accurate understanding of all of the concepts and rules involved [2], [3]. Programming Language courses have high failure rates according to [4-6]. The departments that incorporate this course must satisfy the requirements of the new era and emerging technologies [9]. Accordingly, determining why undergraduate students have poor skills in programming is significantly important [10]. The most important problems encountered by students in learning a programming language (PL) are the practical part, which involve the need for them to practice extensively to achieve higher programming skills [11] and acquire the latest developments in a programming language curriculum such as Java [12]. [13] stated that the difficulty of teaching introductory programming results in higher failure and dropout rates in such courses. Students dropped, failed, or withdrew from the course at rates of between 35% and 50% [14], [15]. Novice students have difficulty learning programming.

Several researchers [1], [10], [16-32] have tried to solve these issues by employing different methods, such as the Web-Based Java Programming Language, 3D animation, a mobile learning application, game-based learning, and visualization. But still have a problem as well as no one mentioned about the requirements for learning programming language.

Consequently, several researches have been done on how to make computer programming fun, students motivated and increase students' performance [22], [23]. In 2010 [24] mentioned that reducing the difficulties of students in understanding the concepts and rules of a programming language can enhance their motivation and competency to learn the course. Finally, to learn Programming Language need to determine the basic requirements that should be considered from students. Therefore, the objective of this paper is to discuss the derivation of programming learning requirements that will induce positive effects when applied to a learning programming language course. The summarized group of requirements would then be verified by multi perspectives via qualitative and quantitative methods.

Section 1 of this paper explains a brief description of the major focus of this study, and describes the objectives and how they are achieved. Section 2 summarizes the challenges and issue based on previous work. Section 3 summarizes and critically analyzes related works. Section 4 describes the methods used in this study. Section 5 presents the analysis of results and discusses the results based on multiple perspectives, and finally, the conclusion of this study is presented in Section 6.

2. PROGRAMMING LANGUAGE CHALLENGES AND ISSUES

Students have to practice and cultivate a strong practical ability to achieve excellent programming skills [11]. The author proposed that, to solve the issue above, students must spend at least 62 hours for JSP technology, 85 hours for Java programming, and 85 hours for J2EE. The method proposed by Wei in 2010, only increases the hours of teaching, which thus leads to feelings of boredom in the student because of the absence of fun or interesting elements that would make learning a programming language easy and interesting to students. Meanwhile, in teaching, [12], [34] used Pair Programming to improve learning of programming skills, and practice in the field of computing. However, the Pair Programming method introduced by the researchers only involves the exchange of knowledge between two students, and this is insufficient because the method depends on the participant’s background. Additionally, Wei (2010) also suggested that practical computer courses be enhanced to keep up with the use of modem teaching methods and individuals. Besides that, students find it difficult to schedule meetings with their partners because of conflicting class and work schedules or family obligations [35], [36].

According to [16], [17] memorizing reserved words in code writing is the most common problem old and novice students face when learning a variety of Java programming levels. Funabiki et al. (2012) solved the code writing problem by developing a web-based Java Programming Learning Assistant System (JPLAS) to help students memorize the reserved words easily as well as to support students by solving the code writing problem and fill-in-the-blanks problems. The results of the web-based JPLAS supported the effectiveness of learning basic Java programming only, so it cannot be applied (JPLAS) for advanced level Java programming courses. Meanwhile, [37] used www.problets.org to improve student comprehension of programming constructs, especially targeting female students, and their coding skills. He found that www.problets.org could improve student knowledge on programming and their ability to write codes. However, problets.org helps females first and foremost. Besides that, the author did not divulge any details as to the method employed in his website. There is only a short workshop description that mentions, *“using Problets.org for problem-solving exercises in introductory c++/java/c# courses”* on Page 1.

According to [38], the problem with code writing is that the syntax of programming languages can be frustrating for students who are new to programming. To gain confidence in understanding programming, Daly [34] compared the results of using pre- and post-test for Alice as a precursor to Java programming and using pure Java programming. The results indicate that the Alice/Java group achieved significantly greater improvement from the pre-test to post-test than the pure Java group in the following areas: objects, classes, methods, parameters, arrays, and variables. However, Daly [34] used only 11 participants for his experiment group and 18 for the control group, so this is not enough for real testing and there is no reference to justify this number. Also, he mentioned that pre and post-tests were used as a survey instrument to only obtain feedback, and this is not enough to validate or justify the results of his paper.

Chang [39] mentioned that difficulties faced in learning the concepts and rules for understanding the Java language must be reduced to enhance the motivation and understanding of students. This process can be done through web-based multiplayer online role-playing game applications that would increase student learning activity by making it fun and interesting to learn game-based contents. He found that the students could have fun while doing their homework and exercises. He also mentioned about the gaps in his study as follows: (1) the most important issue is the integration of an online judge system, which the study did not achieve; (2) Chang [35] aimed to formally evaluate the game in undergraduate level Java programming courses, so as to analyze the possible influences from factors such as gender, culture, age, and country in future work.

For novice students that have difficulties in understanding the basic principles of the Java programming language, Tigrek and Obadat [19] found that applying mobile learning applications (Android) in programming courses could increase engagement in the classroom. However, Tigrek & Obadat used the

Pair Programming method, which only involves the exchange of knowledge between two students, and this is insufficient because the method depends on the participant's background. Besides that, Wei (2010) suggested that practical computer courses be enhanced to keep up with the use of modern teaching methods and individuals. Additionally, the most important drawback of Pair Programming is unbalanced student participation [40]. From examining student experiences with Pair Programming, the majority of them report scheduling conflicts as a major drawback to the program [41]. In addition, to measure knowledge properly, a test group of questions as a pre- and post-test should be conducted and not only based on knowledge of the questions. Meanwhile, in mobile learning, [42] states that, *"When learning programming languages, for instance, it is imperative to work with large screens and comfortable keyboards. It would be a noteworthy achievement to be able to offer a nice programming environment through small screens."* Page 91. However, their research did not measure the effectiveness of mobile learning on students—only a usability measure was used.

Vrachnos and Jimoyiannis [43] used a web-based framework in a Dynamic Algorithm Visualization Environment (DAVE) to solve programming problems for secondary education students learning basic algorithms. The results obtained from an evaluation study provided evidence of the usability of the system and its potential to support student development using efficient mental models involving basic array algorithms. However, this framework is designed only for the array part of a programming course, which is a big gap in this research area because frameworks should be suitable for all kinds of programming concepts and programming languages. Besides that, DAVE was designed for high school students only. Additionally, [44] used the concept visualization technique to address the issue of coding faced by novice students in programming courses and their results confirmed student interest using the technique. However, their research must also include the educational effectiveness of their work via pre and post-test, which they did not do. Plus, they only focused on coding issues for novice students when it is important to increase the understanding of novice students regarding the concepts first and then how to use them followed by teaching them the actual coding.

According to Zhi [45], *"Java is difficult and this reflects the students' general fear of learning Java."* Page 482. Since Java is an object-oriented language, students must think of class and objects. Zhi (2012) integrated theory and practice to concentrate on practical ability by encouraging students to participate in Java programming language related certifications. The author found success in two programming competitions; the first one was a great success, and the second was popular with students.

Learning object-oriented programming is the primary issue for students. Sivasakthi and Rajendran [46] used a questionnaire to analyze student perspective on learning difficult topics of the Java language such as Concurrent programming, User Interface Components with swings, Generic programming, Exceptions and Assertions, Event Handling, Interfaces and inner classes, Graphics programming, Object-Oriented Access controls, Object Orientation, and Fundamental programming structure in Java. In addition, Tan, et al. [10] employed a questionnaire to analyze the student perspective on learning difficult Java programming language topics and solutions. They found that game-based learning as an alternative to teaching and learning computer programming subjects would be a better solution. Therefore, the authors proposed a game-based learning framework to solve the problem. However, they used participants that were mainly beginners in learning programming, so this would lead to a different result if advanced programmers were to take part in the survey. Besides that, they also mentioned that the undergraduates were more interested, motivated, and thought it fun to learn programming, but no detail was given to prove this statement.

Dai, et al. [1] asserts that most of the studies focus on student problems that concentrate on trivial details instead of the essential ones. For example, students will take note of using the String.length() method to get the length of the String object, while the length of the array is obtained via the property of Array.length. Robocode and Karel World used educational games to make the Java language more interesting for students and to focus on essential details during the study. As a result of their study, the Programming Language course in Fudan University received the excellent course award in 2009. However, Dai et al. [1] used Robocode, which is designed according to an object-oriented paradigm, where the packages such as robocode, robocode.control, and robocode.util are provided, including an event interface like robocode.Event. However, Karel World is developed somewhat in depth, and does not aim for the students to learn fundamentals from the beginning.

Unfortunately, many teaching materials used in the university ignore the essential issues in programming courses, which has led to bad coding habits that have caused students to unconsciously write ugly codes from the beginning, such as simple and ambiguous variable and function names, or the use of procedural programming as a pathway to object-oriented concepts. [47] thus offered their opinion on how to solve the problems, i.e. by applying the basic concepts and skills to a programming language, such as the learner must be familiar with the Java programming language, read documentation and code specifications, and have debugging skills. In additionally, [48] talked about the difficulty in understanding debugging in their study and used jGRASP visualization to allow students to understand the Java course more easily as well as create program visualization by dragging. However, the authors did not include any results or details about this method, nor its evaluation and implementation. Furthermore, Zhou [49] analyzed the common problems in learning a programming langugae from two aspects, namely "contents" (teaching contents) and "reveals" (how to layout and display contents), whereby "contents" should be the core aspect of a courseware and "reveals" should assist "contents". The author further proposed a Unified framework of the courseware in programming courses that would help organize contents to make understanding easy for students. However, this framework organizes the contents of certain multimedia courseware only, and as of now there have been no concrete results.

Novice students face difficulty in learning a programming language. To make learning for novice students easier, [50] used Python and Visual Logic. They found that Python is suitable for novice Computer Science students and that Python makes it easier for them to learn programming. However, Python cannot be adapted to learning next-level programming such as object-oriented programming.

3. PROGRAMMING LEARNING REQUIREMENTS

This section discusses the requirements of learning a programming language based on previous work. First of all, there are only a few papers that implicitly talk about the requirements for learning programming language and these are elaborated as follows:

Ziafati, et al. [51] presented four requirements for a DBI (Desire-Belief-Intention)-based programming language, but this study was concerned with two requirements only (Real-time reaction and response to events) in facilitating the implementation of the system. On the other hand, Masterson [52] mentioned two main requirements for students to learn a programming language i.e. simplicity and power.

> *"The first requirement, simplicity, refers to the ease with which students can learn a language, at least to the degree that they can use it to solve simple problems. A simple language can be learned quickly, leaving more time for students to apply the language to the subject matter of a course. The second requirement, power, is a measure of the ease with which a programming language can be applied to complicated problems." Page 181*

Additionally, Masterson [52] mentioned cognitive efficiency as a third requirement that facilitates students when thinking about problems.

> *"Cognitive efficiency is closely related to the requirements of simplicity and power. Indeed, ease of learning, and especially ease of application, imply a rich notation for representing problems in various domains." Page 181 and 182.*

Wei [11] mentioned that students have to practice with strong practical ability to master JAVA program development skills. Table 1 summarizes the programming learning requirements based on previous research.

Table 1. Programming Learning Requirements based on pPevious Research

No	Programming learning requirements	References
1	Real-time reaction Response to events	[51]
2	Simplicity Power Cognitive efficiency	[52]
3	Thinking skill and Practices	[11]

4. METHODS AND MATERIALS

This study was carried out via two kinds of research methods i.e. the qualitative and quantitative methods. This study used an interview approach to assess the perspective of experts. To investigate the perspectives of real users (students), a survey approach using instruments such as questionnaires were used. This is further elaborated upon in the subsequent section.

A qualitative method via an interview approach was conducted at the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM). The purpose of the interview was to verify the programming learning requirements based on previous work. The interview began with an explanation of the purpose of the research and why the interview method was used. A total of 5 experts were interviewed based on [53-55]. Each interview took 1-2 hours, during which all requirements were asked, as suggested by [26], [27], [11].

A quantitative method using a survey approach was employed. The questionnaire began with an explanation of the purpose of the investigation and why the questionnaire method was used. The purpose of the questionnaire was to gain feedback about the requirements from real users. A total of 30 participants (30 undergraduate/ postgraduate students from Universiti Kebangsaan Malaysia, Universiti Putra Malaysia, and Universiti Tenaga Nasional, aged between 21 to 31 years) were invited to answer the questionnaire. All the students were enrolled in Semesters 2 (2013-2014 session), semester 1/2 (2014-2015 session), and semester 1 (2015-2016 session).

All questions, which were based on previous studies and interviews [56], [58]. Prior to answering the questionnaire. During the survey, each respondent is required to write a comment or suggestion about the requirements of learning programming language towards the enhancement of student learning.

4.1. Data Collection

Interviews were conducted in person as a face to face interview in UKM. The interview questions was structured and adopted from previous work, the details of interview questions as shown in Table 2. Survey was in this study conducted using questionnaire likert scale, distributed by self-administration to students. The questions of survey are based on the outcomes of interviews. The main purpose that leads this study to conducted two perspectives of expert and students (undergraduate and postgraduate) was to verify by existing generation and to compare between their viewpoints of the same elements.

Table 2. Interview Questions

No	Questions
1	Based on your experiences, what is the requirement to learn programming language?
2	Do you think the requirements can support the learning? In which metrics?
3	I summarized a group of requirements based on previous work; can specify which one is the most suitable for learning programming language?
4	In your opinion, what is the impact of these requirements on learning?
5	Based on your experiences, how these requirements increase the effectiveness, level of interest and motivation for students in learning programming language?

4.2. Survey Instrument

The instrument in this research using 5-points likert scales to measure the level of agreement namely, strongly disagree, disagree, quiet agree, agree, and strongly agree. For each question, the students were required to select one scale that reflects their belief. The questions of survey is based on previous work and the interview schedule are presented in Tables 1 and 2. The purpose of survey is to measure the level of requirements agreement based on real users.

5. RESULTS AND ANALYSIS

The results and analyses of both approaches are presented.

5.1. Interview

The results of the analysis revealed that the interviewed participants gave positive answers in support of, and to solidify any issues; a few examples are illustrated below:

> Interviewee 1:
> *"The most impact requirement of learning programming language is student motivation. How to motivate students such as through game or game elements or any facility led to increase the student interest to learn this subject…"*

> Interviewee 2:
> *"Each subject has many requirements such as for programming language, all student must practice as much as passable in order to memorise the reserved word that dayly used. How to motivate studeny is an impotant requirement for all subjects…"*

Interviewee 3:
“I believe if students followed all requirements based on their lecturers, they will gain the effectiveness, motivation, engagement and interesting in all subjects…”

This section analyzes the information provided by the interviewed (lecturers and teachers) to verify programming learning requirements. As the below:

1. In general all interview with (lecturers and teachers) was provided agree feedback for all previous programming learning requirements.
2. They are suggesting more requirements for learning programming language such as (Thinking skill: means to enhance thinking skill by making students working on teams; it’s very important for new student to learn the concepts of programming language, that is mean to make a strong base for their knowledge in future when they want to adopt with other programming language; Motivation through game rule, use attractive things in order to increase the interesting of learning programming language during class or through application; Motivation student through the fun of game elements: means use drag-drop questions or Multiple choice questions (MCQ) that should be easy and understandable from students then the fun of challenge among student when show their result; and References: means students must keep with the latest update references (book).

Finally, in subsequent section the result of the survey is presented.

5.2. Survey

This section analyzes the information provided by graduate students and postgraduate students to verify programming learning requirements, as shown in Table 3.

Table 3 The Mean Score of Acceptance for Programming Learning Requirements

Programming Learning Requirements	Experts (N=5)		Undergraduate/ Postgraduate students (N=30)	
	Mean	S.D	Mean	S.D
Real-time reaction;	4.40	.894	4.60	.675
Response to events;	4.20	.837	4.57	.679
Simplicity;	4.20	.837	4.67	.606
Power;	4.40	.894	4.63	.615
Cognitive efficiency;	4.00	1.000	4.50	.731
Practices;	4.20	.837	4.57	.679
Motivation through game rules;	4.00	1.000	4.60	.621
Used update references and basic concepts;	4.00	1.000	4.67	.606
Motivation student through the fun of game elements.	4.00	1.000	4.57	.679

According to the result in Table 3, the minimum value of mean based on experts’ perspective is 4.00, but 4.50 based on students’ perspective. It shows respondents agree all programming requirements in Table 3. The first requirement is “Real-time reaction” has 4.40 from experts’ perspective and 4.60 from students’ perspective. But in “motivation student through the fun of game elements” have 4.00 from experts’ perspective and 4.57 from students’ perspective. The standard deviation S.D values in over all requirements are less than 1.00, that means the amount of variation or dispersion of a set of data values are adequate.

In conclusion all programming learning requirements based on previous research and interview outcomes indicate positive acceptance from the two perspectives. Some suggestion to improve programming learning are students should practice as much as possible; provide more examples with the solutions; student must be exposed to different assessment types and finally, provide students with fun and enjoyable learning environments..

6. CONCLUSION

This study attempts to identify the programming learning requirements based on previous work. The identified requirements are then verified by experts and students. A set of learning programming language requirements are identified such as (1) Real-time reaction; (2) Response to events; (3) Simplicity; Power; (4) Cognitive efficiency; (5) Practices; (6) Motivation through game rules ; (7) Used update references and basic concepts; and (9) Motivation student through the fun of game elements. Finally, the future work would match these requirements with the enjoyable environments such as gamification technique in order to measure the effectiveness on students learning.

ACKNOWLEDGEMENTS

This work is supported by Centre for Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Multimedia Software and Usability Research Group (DPP-2015-022) and FRGS/2/2014/ICT05/UKM/02/1 research grant.

REFERENCES

[1] K. Dai, Y. Zhao, and R. Chen, *"Research and Practice on Constructing the Course of Programming Language,"* in Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on, 2010, pp. 2033-2038.

[2] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," in *ACM SIGCSE Bulletin*, 2005, pp. 14-18.

[3] J. M. Rodríguez Corral, A. Civit Balcells, A. Morgado Estévez, G. Jiménez Moreno, M. J. Ferreiro Ramos, "A Game-based Approach to the Teaching of Object-Oriented Programming Languages," *Computers & Education*, vol. 73, pp. 83-92, 4// 2014.

[4] A. Robins, J. Rountree, N. Rountree, "Learning and Teaching Programming: A Review and Discussion," *Computer Science Education*, vol. 13, pp. 137-172, 2003.

[5] A. Adorjan, I. Friss de Kereki, *"Multiple Intelligence Approach and Competencies Applied to Computer Science 1,"* in Frontiers in Education Conference, 2013 IEEE, 2013, pp. 1170-1172.

[6] P. Kinnunen, L. Malmi, *"Why Students drop out CS1 Course?,"* in Proceedings of the second international workshop on Computing education research, 2006, pp. 97-108.

[7] J. Bennedsen, M. E. Caspersen, "Failure Rates in Introductory Programming*," ACM SIGCSE Bulletin*, vol. 39, pp. 32-36, 2007.

[8] A. P. Ambrósio, F. M. Costa, L. Almeida, A. Franco, J. Macedo, *"Identifying Cognitive Abilities to Improve CS1 Outcome,"* in Frontiers in Education Conference (FIE), 2011, 2011, pp. F3G-1-F3G-7.

[9] Z. Teng, *"Reform and Practice of J2EE Course,"* in Computer Science & Education (ICCSE), 2013 8th International Conference on, 2013, pp. 1272-1276.

[10] P.-H. Tan, C.-Y. Ting, S.-W. Ling, *"Learning difficulties in Programming Courses: Undergraduates' Perspective and Perception,"* in Computer Technology and Development, 2009. ICCTD'09. International Conference on, 2009, pp. 42-46.

[11] X. Wei, *"Research of Practical Course Teaching of JAVA Language,"* in 2010 International Conference on Educational and Information Technology (IEEE), 2010.

[12] L. Han, X. Wenjuan, *"An Experimental Research of the Pair Programming in Java Programming Course,"* in e-Education, Entertainment and e-Management (ICEEE), 2011 International Conference on, 2011, pp. 257-260.

[13] W. Pullan, S. Drew, S. Tucker, *"An Integrated Approach to Teaching Introductory Programming,"* in e-Learning and e-Technologies in Education (ICEEE), 2013 Second International Conference on, 2013, pp. 81-86.

[14] P. J. Denning, "The Field of Programmers myth*," Communications of the ACM*, vol. 47, pp. 15-20, 2004.

[15] J. Flieger, J. D. Palmer, "Supporting Pair Programming with JavaGrinder*," Journal of Computing Sciences in Colleges*, vol. 26, pp. 63-70, 2010.

[16] N. Funabikiy, Y. Korenaga, Y. Matsushima, T. Nakanishi, K. Watanabe, *"An Online Fill-in-the-Blank Problem Function for Learning Reserved Words in Java Programming Education,"* in Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on, 2012, pp. 375-380.

[17] N. Funabiki, Y. Korenaga, T. Nakanishi, K. Watanabe, *"An Extension of Fill-in-the-Blank Problem Function in Java Programming Learning Assistant System,"* in Humanitarian Technology Conference (R10-HTC), 2013 IEEE Region 10, 2013, pp. 85-90.

[18] N. Funabiki, Y. Matsushima, T. Nakanishi, K. Watanabe, N. Amano, "A Java Programming Learning Assistant System Using Test-Driven Development Method*," IAENG International Journal of Computer Science*, vol. 40, 2013.

[19] S. Tigrek, M. Obadat, *"Teaching Smartphones Programming using (Android Java): Pedagogy and Innovation,"* in Information Technology Based Higher Education and Training (ITHET), 2012 International Conference on, 2012, pp. 1-7.

[20] S. Heckman, T. B. Horton, M. Sherriff, *"Teaching Second-Level Java and software Engineering with Android,"* in Software Engineering Education and Training (CSEE&T), 2011 24th IEEE-CS Conference on, 2011, pp. 540-542.

[21] J. J. Magee, L. Han, *"Integrating a Science Perspective into an Introductory Computer Science Course,"* in Integrated STEM Education Conference (ISEC), 2013 IEEE, 2013, pp. 1-4.

[22] H. Tsukamoto, N. Nitta, Y. Takemura, H. Nagumo, *"Work in Progress: Analysis of the Relationship between Teaching Contents and Motivation in Programming Education,"* in Proceedings of the 2012 IEEE Frontiers in Education Conference (FIE), 2012, pp. 1-2.

[23] S. Montero, P. Díaz, D. Díez, I. Aedo, "Dual Instructional Support Materials for Introductory Object-Oriented Programming: Classes vs. Objects," in *Education Engineering (EDUCON),* 2010 IEEE, 2010, pp. 1929-1934.

[24] M. Chang, *"Web-Based Multiplayer Online Role Playing Game (MORPG) for Assessing Students' Java Programming Knowledge and Skills,"* in Digital Game and Intelligent Toy Enhanced Learning (DIGITEL), 2010 Third IEEE International Conference on IEEE, 2010, pp. 103-107.

[25] F. L. Khaleel, N. S. Ashaari, T. S. Meriam, T. Wook, A. Ismail, *"The Study of Gamification Application Architecture for Programming Language Course,"* in Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication, 2015a, p. 17.

[26] F. L. Khaleel, N. S. Ashaari, T. S. Meriam, T. Wook, A. Ismail, *"User-Enjoyable Learning Environment Based on Gamification Elements,"* in International Conference on Computer, Communication, and Control Technology (I4CT 2015),, Kuching, Sarawak, Malaysia, 2015b, p. 221.

[27] F. L. Khaleel, N. S. Ashaari, T. S. Meriam, T. Wook, A. Ismail, "The Architecture of Dynamic Gamification Elements Based Learning Content*," Journal of Convergence Information Technology*, vol. 11, pp. 164-177, 2016a.

[28] F. L. Khaleel, T. S. M. T. Wook, N. S. Ashaari, A. Ismail, "Gamification Elements for Learning Applications*," International Journal on Advanced Science, Engineering and Information Technology*, vol. 6, p. 1-8, 2016b.

[29] M. Abdurohman, A. Sasongko, "Software for Simplifying Embedded System Design Based on Event-Driven Method*," International Journal of Electrical and Computer Engineering*, vol. 5, p. 491, 2015.

[30] L. Elaachak, A. Belahbibe, M. Bouhorma, "Towards a System of Guidance, Assistance and Learning Analytics Based on Multi Agent System Applied on Serious Games*," International Journal of Electrical and Computer Engineering*, vol. 5, p. 344, 2015.

[31] A. Alali, I. Assayad, M. Sadik, "Multilevel MPSoC Performance Evaluation: New ISSPT Model*," International Journal of Electrical and Computer Engineering*, vol. 5, 2015.

[32] S. Q. Ameen, R. C. Muniyandi, "Improvement at Network Planning Using Heuristic Algorithm to Minimize Cost of Distance between Nodes in Wireless Mesh Networks*," International Journal of Electrical and Computer Engineering*, pp. 1-8, Just accepted, 2016.

[33] Q. Mahmoud, M. A. Talib, Z. Maamar, E. Bataineh, L. Jololian, *"Revitalizing the introductory programming course,"* in GCC Conference and Exhibition (GCC), 2011 IEEE, 2011, pp. 545-548.

[34] P. Gayathri, "Improving Student's Programming Ability using Peer Learning Agent*," Middle-East Journal of Scientific Research*, vol. 20, pp. 2294-2296, 2014.

[35] B. Hanks, "Student Performance in CS1 with Distributed Pair Programming," in *ACM SIGCSE Bulletin*, 2005, pp. 316-320.

[36] B. Hanks, "Empirical Evaluation of Distributed Pair Programming*," International Journal of Human-Computer Studies*, vol. 66, pp. 530-544, 2008.

[37] A. N. Kumar, *"Using Problets for Problem-Solving Exercises in Introductory C++/Java/C# Courses,"* in Frontiers in Education Conference, 2013 IEEE, 2013, pp. 9-10.

[38] T. Daly, "Minimizing to Maximize: An Initial Attempt at Teaching Introductory Programming using Alice*," Journal of Computing Sciences in Colleges*, vol. 26, pp. 23-30, 2011.

[39] M. Chang, *"Web-Based Multiplayer Online Role Playing Game (MORPG) for Assessing Students' Java Programming Knowledge and Skills,"* in Digital Game and Intelligent Toy Enhanced Learning (DIGITEL), 2010 Third IEEE International Conference., 2010, pp. 103-107.

[40] D. Tsompanoudi, M. Satratzemi, S. Xinogalos, "Distributed Pair Programming Using Collaboration Scripts: An Educational System and Initial Results," *Informatics in Education*, vol. 14, 2015.

[41] S. Faja, "Pair Programming as a Team based Learning Activity: A Review of Research," *Issues in Information Systems*, vol. 12, pp. 207-216, 2011.

[42] O. Ortiz, P. M. Alcover, F. Sanchez, J. A. Pastor, R. Herrero, "M-Learning Tools: The Development of Programming Skills in Engineering Degrees," Tecnologias del Aprendizaje, IEEE Revista Iberoamericana de, vol. 10, pp. 86-91, 2015.

[43] E. Vrachnos, A. Jimoyiannis, "Design and Evaluation of a Web-based Dynamic Algorithm Visualization Environment for Novices," *Procedia Computer Science*, vol. 27, pp. 229-239, 2014.

[44] C. Watson, F. W. Li, R. W. Lau, "Learning Programming Languages through Corrective Feedback and Concept Visualisation," in Advances in Web-Based Learning-ICWL 2011, ed: Springer, 2011, pp. 11-20.

[45] Y. Zhi, "Investigation and Implementation of Teaching Methods of Java Language," in *Computer Science and Electronics Engineering (ICCSEE),* 2012 International Conference on, 2012, pp. 482-485.

[46] M. Sivasakthi, R. Rajendran, "Learning difficulties of'Object-Oriented Programming Paradigm using Java': Students' Perspective," *Indian Journal of Science & Technology*, vol. 4, 2011.

[47] P. Yu, L. Yang, "Programming Skills Training in Programming Language Courses," in *Educational and Information Technology (ICEIT),* 2010 International Conference on, 2010, pp. V3-14-V3-16.

[48] I. Cross, H. James, "Using the new jGRASP Canvas of Dynamic Viewers for Program Understanding and Debugging in Java Courses," *Journal of Computing Sciences in Colleges*, vol. 29, pp. 37-39, 2013.

[49] X. Zhou, "Research on the CAI Software for Teaching Java," in *Education Technology and Computer Science (ETCS),* 2010 Second International Workshop on, 2010, pp. 787-789.

[50] K. K. Agarwal, A. Agarwal, L. Fife, "Python and Visual Logic: A Good Combination for CS0," *Journal of Computing Sciences in Colleges*, vol. 27, pp. 22-27, 2012.

[51] P. Ziafati, M. Dastani, J.-J. Meyer, L. van der Torre, "Agent Programming Languages Requirements for Programming Autonomous Robots," in Programming Multi-Agent Systems, ed: Springer, 2013, pp. 35-53.

[52] F. Masterson, "Evaluating Logo: A Case Study in Requirements for Student Programming Languages," *Computers in the Schools*, vol. 2, pp. 179-195, 1985.

[53] J. Nielsen, Usability engineering: Elsevier, 1994.

[54] J. W. Creswell, Qualitative inquiry and research design: Choosing among five approaches: Sage, 2007.

[55] J. W. Creswell, Qualitative inquiry and research design: Choosing among five approaches: Sage, 2012.

[56] N. A. Matzin and W. SengYue, "Design and evaluation of history digital game based learning (DGBL) software," Journal of Next Generation Information Technology, vol. 4, p. 9, 2013.

[57] N. A. Matzin, W. S. Yue, “History Educational Games Design,” in *Electrical Engineering and Informatics,* 2009. ICEEI'09. International Conference on, 2009, pp. 269-275.

[58] F. L. Khaleel, “Recruitment and Job Search Application,” M.Sc. IT. thesis, Universiti Utara Malaysia, 2011.