academicJournals

Vol. 11(18), pp. 1757-1765, 23 September, 2016
DOI: 10.5897/ERR2016.2948
Article Number: 70ED17760719
ISSN 1990-3839

http://www.academicjournals.org/ERR

Educational Research and Reviews

Full Length Research Paper

The use of engineering design concept for computer programming course: A model of blended learning environment

Kasame Tritrakan*, Pachoen Kidrakarn and Manit Asanok

Faculty of Education, Mahasarakham University, Thailand.

Received 28 July, 2016; Accepted 13 September, 2016

The aim of this research is to develop a learning model which blends factors from learning environment and engineering design concept for learning in computer programming course. The usage of the model was also analyzed. This study presents the design, implementation, and evaluation of the model. The research methodology is divided into three phases, they are: reviewing related literatures and surveying needs and problems in teaching computer programming in order to analyze and synthesize the elements of the model, developing a model which blends learning environment and learning activities based on engineering design processes, and evaluating the effects of using implemented environment on the learners' programming conceptual understanding, problem-solving using programming skills, program analytical skills, and attitude in learning programming. The samples in this study include 8 experts who examine the tentative model and 52 undergraduate students of computer science program in Bansomdejchaopraya Rajabhat University, Thailand, to evaluate the implemented model. The student samples were separated into two groups which are control and experiment group. The results of the comparison between pre-test and post-test scores showed that the programming conceptual understanding, problem-solving by using programming skills, and program analytical skills of the students were significantly increased. In addition, in comparing between control and experiment groups, the scores of problem-solving using programming skills and program analytical skills of the experiment group are significantly higher than that of the control group. However, there were no significant differences in the scores of programming conceptual understanding between those groups. It was also found that the students in experiment group have a high attitude in learning programming in high level.

Key words: Blended learning environment, engineering design concept, computer programming.

INTRODUCTION

It is generally accepted that computer programming is difficult for both the learner and teacher (Allison et al., 2002; Jenkins, 2002). This is because it requires more of individual's skills than knowledge (Sarpong, 2013; Hadjerrouit, 2008; Miliszewska and Tan, 2007).

Currently, it is observed that there has not been much

*Corresponding author. E-mail: pui043@gmail.com.

success in teaching programming, as many programming students do not perform up to expectation, and as a result, many of them drop out in many countries (Denning and McGettrick, 2005) and also in Thailand (Sankas, 2010). Thus, it is of utmost importance to use different methods and technology to improve students' competence.

Generally in programming courses, learners are first introduced to the programming conceptual knowledge which is about the language features such as syntax, variables, control structures, and functions. Knowledge of the language features is necessary to use the programming language. However, only this knowledge is not sufficient to achieve competence in programming (Kwon et al., 2011).

After teaching language features, the lecturer shows examples of coded program to demonstrate how the instructions are coded and how they work. In order to understand how the programs work and what the output of running the programs is, the program analytical skills are needed to analyze the results of processes and relationship between each line of code. After acquiring the concept of programming, the students are given problems to solve.

In this phase, the learners have to use the problem-solving skills to create the solution and convert into program instruction codes. Lack of program analytical skills and problem-solving by using programming skills are viewed as the main cause of failure in programming learning (Jenkins, 2002; Ismail et al., 2010; Robins et al., 2003). Furthermore, it is also important for students to be highly motivated in order to be successful in learning programming (Jenkins, 2001).

Blended learning (BL)

Blended learning is an education model that combines different types of learning strategies (Bonk and Graham, 2006). In this study, it refers to a combination of online and traditional face-to-face learning. In this, students learn new concepts on their own through the internet and the teachers support individual students who need extra attention. As a result, in the classroom, the teachers can focus on giving instructions that help students improve their higher skills. This combination approach has been adopted in many university courses because it ensures an effective learning environment for students. It combines the advantages and disadvantage of traditional face-to-face education and e-learning. The learners can improve their skills and knowledge in the course (Alducin-Ochoa and Vazquez-Martinez, 2016; Ryberg and Dirckinck-Holmfeld, 2010). There are lots of researchers which aim to harness the efficiency of blended learning approach to computer programming course (Djenic et al., 2011; Deperlioglu and Kose, 2010; Abbas et al., 2009; Hadjerrouit, 2008; Boylea et al., 2003). The results of those studies have shown that BL approach can efficiently improve the learners' programming competency. Furthermore, the Thai government has a policy B.E.2558 "Reduce instructional time, increase learning time", so the researchers considered that BL can answer this policy.

Engineering design concept (EDC)

Engineering design is the systematic and adaptable processes for an engineer in solving the problem (Dym et al., 2005). It contains many steps from defining to solve the problem which is expanded on the traditional role of Problem-Based Learning (PBL) (Strobel and Carr, 2011). The engineering design processes include (Asunda and Hill, 2007):

1. Identify the problem
2. Research the problem
3. Develop the possible solutions
4. Select the best solution
5. Construct the prototype
6. Test and evaluate the solution
7. Communicate the solution and
8. Redesign.

There are some researches which applied the engineering design concept to education especially STEM to create higher skills such as critical thinking skills and problem-solving skills (Evans et al., 2016; Dixon and Johnson, 2012; Strobel and Carr, 2011; Robert et al., 2009). With the earlier mentioned reasons, the researchers considered combining the engineering design concept with the blended learning to the model to enhance the analytical skills and problem-solving skills in programming.

Research aims

The purpose of this research was to develop blended learning environment model (BLE) using engineering design concept learning activities to computer programming courses. In addition, this research investigated the effects of using this developed model in programming conceptual understanding, problem-solving using programming skill, program analytical skill, and attitude towards learning programming.

METHODOLOGY

This research was conducted in three phases:

The 1st phase: To review related literature and to analyze and

synthesize the elements of the model. There were 3 steps included in this:

Step 1: Search and review the principles, theories, and previous research to analyze and synthesize the components of the model. In this step, the principles of a learning environment, blended learning, learning motivation, engineering design concept, learning computer programming, and desired characteristics of students for learning programming were studied.

Step 2: Survey the computer programming lecturers' challenges and needs. The 57 lecturers who taught computer programming courses in Rajabhat Universities of Thailand were selected to answer the questionnaire. The questions covered challenges and needs faced in learning environment and teaching. The responses were gathered and analyzed using mean and standard deviation.

Step 3: In-depth face-to-face interview 5 programming specialists through the programming learning environment, teaching methods and desire characteristics of learners in learning programming. The structured interview forms were adopted. The results were clustered, relation analyzed, and presented in descriptive format.

The 2nd phase: To develop the model of blended learning environment. There were 3 steps including:

Step 1: Synthesize and design the tentative model using information from the previous phase based on the conceptual framework of systematic development approach (von Bertalanffy, 1968) which includes input, process, output, and feedback.

Step 2: Evaluate appropriateness of the components and model confirmation. In this step, 8 experts who are skilled in programming teaching and educational technology were selected to assess the tentative model. The 1-5 rating scale was conducted and used to evaluate the tentative model.

Step 3: Develop the learning environment and teaching activities based on the developed model. Both offline and online learning environment were established. The teaching activities which use engineering design concept was also introduced in the form of course syllabus and teaching plan. After developing, 3 experienced programming lecturers were selected to investigate the suitability of developed learning environment and teaching method. In this step, the tools to investigate the desired characteristics of students for learning programming were also inspected and approved by IOC and try out.

The 3rd phase: Study the effect of using the implemented environment and teaching activities. There were 5 steps included:

Step 1: Assess the desired characteristics of the student before the course (pre-test).

Step 2: Setup the learning environment and teaching equipment. To study the effects of the designed model, a total of 52 students who attended Web Programming class in Computer Science at Bansomdejchaopraya Rajabhat University in Thailand took part in this study. The students were formed into 2 groups, 25 students in the experiment group and 27 students in the control group by random selection.

Step 3: Perform teaching and learning. The experiment group took the course via the developed model while the control group was taught with the traditional face-to-face teaching method.

Step 4: Assess the competence of the students after the course (post-test). After the course, both groups were examined by the same final examinations with inspecting tools.

Step 5: Evaluate the results. The results of examinations were analyzed using statistical methods including mean, standard deviation, t-test, and one-way MANOVA.

RESULTS

BLE-EDC Model

The blended learning environment model using engineering design concept, called BLE-EDC model, used the principles of blended learning environment and learning activities based on engineering design concept. In addition, the learning motivation factors were also added into the model to encourage positive learner behavior. The BLE-EDC model is shown in Figure 1. The model aims to develop the competence of programming students which are programming conceptual understanding, program analytical skills, and problem-solving using programming skills. The model consists of 4 types of environment: physical, mental, social, and information environment. The physical environment refers to learning environment that learners can perceive by the 5 senses (sight, hearing, touch, taste, and smell). It includes the environment in class/laboratory room such as computers, table, chair, learning equipment and stationary, temperature, light, and quietness.

1. The mental environment refers to learning environment that learners can perceive in their psyche which affects their impact enthusiasm, attitude, IQ, personality, and teaching method.
2. The social environment refers to the interaction between lecturer to students and students to students.
3. The information environment refers to the storage, retrieval, and transfer of information and knowledge.

Input: Consists of 3 components which are:

1. Blended learning environment which combines the offline and online environments. Offline environment is the environment inside and outside the class/lab room where face-to-face teaching and learning take place. It includes the learning equipment (that is, tables, chairs, whiteboard, books, stationery, audio system, computer PC, and internet connection) and atmosphere (that is, temperature, quietness, neatness, and room size). The online environment is the internet which composes of an online course, tests, news and announcements, consulting, scaffolding, additional resources, and learning management system (LMS).

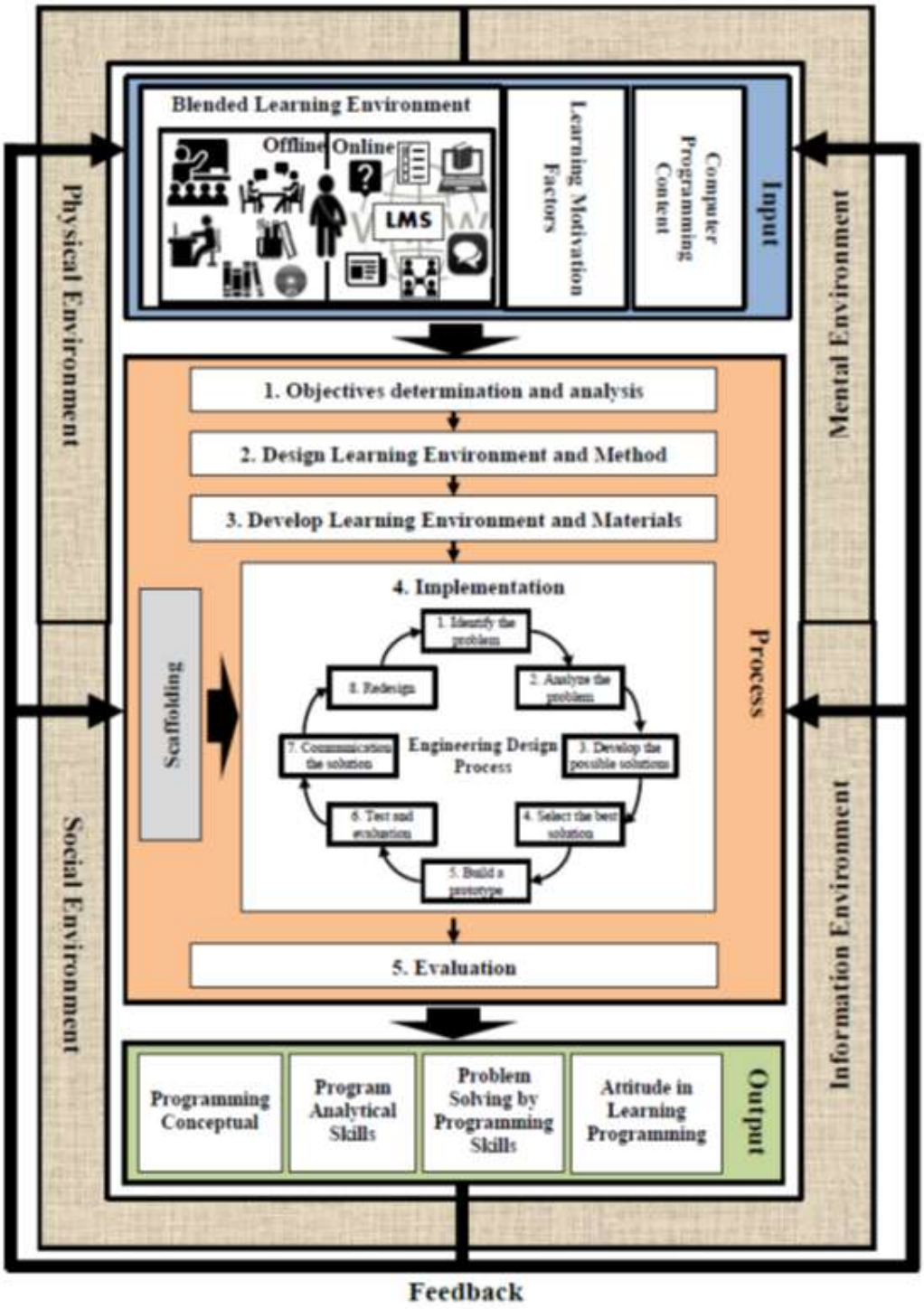

Figure 1. The blended learning environment model using engineering design concept: BLE-EDC model.

2. Learning motivation factors which are the forces which push the students to achieve their learning goals. It is one of the main factors for the accomplishment of programming learning (Jenkins, 2001; Mohorovicic and Strcic, 2011). There are 3 types of motivation for programming learning based on Jenkins (2001). They are

intrinsic, extrinsic, and social motivation. The motivation is given both in face-to-face and online environment. In the online environment, the news on the benefits of being a programmer is present, while in the face-to-face environment, the teacher periodically motivates the students in the classroom.
3. Computer programming content which is the knowledge in fundamental programming and language's syntax and semantics containing the principle of computer programming, introduction to the language, data types, variables, operators, input and output, control structures, functions, and arrays.

Process: Consists of 5 steps which are:

1. Objectives determination and analysis: This step is to study and analyze the problems and needs assessment for determining the objectives and characteristics of designing learning environment.
2. Design learning environment and methods: This step is to plan and define the resources used to build the learning environment and activities.
3. Develop learning environment and materials: This step is to develop the blended learning environment, courseware, and teaching and learning activities. The assessment tools are also conducted in this step.
4. Implementation: In this step, the developed learning environment, program courseware, and its activities are used in computer programming course. The learning activities are based on engineering design processes (Ronald and Strobel, 2011; Robert et al., 2009) which are:

Step 1: Define clearly a specific problem and desired solution.
Step 2: Analyze the problem involved in the input and output variables. Their data types and structures are also assigned.
Step 3: Develop the possible solutions. In this step, possible algorithms are self-developed or searched from other resources.
Step 4: Select the best solution. Learners analyze the advantage/disadvantage of each algorithm and select the best one.
Step 5: Build a prototype program using the specific programming language.
Step 6: Test and evaluate the developed prototype to find bugs and fix them.
Step 7: Communicate and discuss the algorithm and its result with other groups.
Step 8: Improve the program.

5. Evaluation: This step is to evaluate the result of the usage of developed learning environment, courseware, and activities using the assessment tools.

Output: Consists of 4 outcomes which are:

1. Programming conceptual knowledge
2. Program analytical skills
3. Problem-solving by programming skills
4. Attitude in learning programming

Learning and teaching

There are two learning approach in the BLE-EDC model, face-to-face and online learning in 60:40 ratios. In online learning, the students learn the programming conceptual knowledge, which is the surface learning (Mohorovicic and Strcic, 2011), on their own using the online courses and tutorials. In addition, some program analytical skills are also developed by analyzing the example coded programs provided on the online tutorials and coaching by the lecturer. After students have understood the programming conceptual knowledge and have acquired program analytical skills, the next class is held face-to-face learning in the class/lab room. In the classroom, the students are focused on developing the higher skills, program analytical skills and problem-solving by using programming skills, by applying the engineering design concept to their activities to find the solutions to the given problems.

Model usage evaluation

Results of testing

Table 1 shows the scores of programming conceptual understanding, program analytical skills, and problem-solving by using programming skills. The results show that the average scores of post-test are higher than that of the pre-test in both control and experiment group. And all three average post-test scores in the experiment group are higher than the control group.

Comparing between pre-test and post-test

In order to determine that the BLE-EDC model can enhance students' competence in learning programming, the independent t-test analysis between pre-test and post-test are done. Table 2 shows the statistical results between pre-test and post-test of the experiment group. The results show that the average post-test scores of programming conceptual understanding, program analytical skills, and problem-solving by using programming skills are significantly higher than the pre-test scores. These indicate that the BLE-EDC model can improve the knowledge and required programming skills

Table 1 Mean score and standard deviation of programming conceptual understanding, program analytical skills, and problem-solving by using programming skills.

Dependent variable	Control group (N=27)				Experiment group (N=25)			
	Pre-test		Post-test		Pre-test		Post-test	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
Programming conceptual understanding	18.11	5.15	65.15	14.10	18.68	4.49	68.60	13.93
Program analytical skills	6.93	3.20	37.48	13.63	6.52	2.82	48.48	13.60
Problem-solving by using programming skills	2.74	2.18	11.07	6.11	2.88	2.19	20.16	7.15

Table 2. t-test analysis between pre-test and post-test of the experiment group.

Dependent variable	Test	Mean	df	t	Sig.
Programming conceptual understanding	Pre-test Post-test	18.68 68.80	24	19.070	0.000*
Program analytical skills	Pre-test Post-test	6.52 48.48	24	15.385	0.000*
Problem-solving by using programming skills	Pre-test Post-test	2.88 20.16	24	12.224	0.000*

*Correlation is significant at the .05 level.

Table 3. Comparison of the result between the experiment and control group.

Dependent variable	Group	Number of students	Mean	F	Sig.
Programming conceptual understanding	Experiment group Control group	25 27	68.80 65.15	0.881	0.352
Program analytical skills	Experiment group Control group	25 27	48.48 37.48	8.472	0.005*
Problem-solving by programming skills	Experiment group Control group	25 27	20.16 11.07	24.408	0.000*

*Correlation is significant at the .05 level.

of the learners.

Comparing between control and experiment group

To answer the research questions, how the BLE-EDC model affects the students' acquisition of programming conceptual knowledge, program analytical skills, and problem-solving using programming skills, after the courses, both groups took the same final examinations. The examination included 3 parts to test programming conceptual, program analytical skills and problem-solving using programming skills. Table 3 shows the one-way ANOVA results between the experiment and control group. As shown in Table 3, the programming conceptual average scores of the experiment and control group were not significantly different. Nonetheless, the program analytical skills and problem-solving using programming skills average scores of the experiment group were significantly higher than the control group. The experiment results show that the BLE-EDC model can improve the program analytical skills and problem-solving

Table 4. Attitude survey results.

Statement	Number of students giving response as:					Avg.
	1	2	3	4	5	
The computer programing is an interesting course	0	0	2	11	12	4.40
I feel active when in computer programming class	0	0	4	12	9	4.20
think that computer programming course can increase my chance of finding a job	0	0	2	5	18	4.64
Computer programming is an easy subject	0	6	10	9	0	3.12
Computer programming helps me to be accepted by friends, family and social	0	0	15	6	4	3.56
I think that computer programming is an important course	0	1	3	7	14	4.36
I encourage attending a computer programming class	1	0	16	8	0	3.24
I am happy when learning computer programming	0	0	6	12	7	4.04
I feel energetic when the computer programming class is coming	0	0	9	10	6	3.88
I think that computer programming is essential for country development	0	1	2	15	7	4.12
I am enthusiastic to learn computer programming	0	1	6	14	4	3.84
I intend to learn more beyond the class	0	0	10	12	3	3.72
I intend to do the assignment	0	0	7	13	5	3.92
I feel relaxed when in computer programming class	0	1	14	6	4	3.52
I pay attention when learning computer programming	0	1	4	12	8	4.08
Average						**3.91**

using programming skills better than traditional learning. Meanwhile, it is effective in transferring programming conceptual knowledge the same way as the traditional learning.

Attitude study

In this study, the attitude in learning programming was also considered. The survey was done at the end of the course to find out the attitude of students in learning computer programming after learning with BLE-EDC model. A list of 15 statements was examined. The 25 students in the experiment group were asked to express their opinion on the 1-5 Likert scale (1 = strongly disagree, 2 = disagree, 3 = not sure, 4 = agree, 5 = strongly agree). The survey results are shown in Table 4. According to the result in Table 4, the students had a good attitude towards learning computer programming. Mostly, they strongly agree that computer programming course can increase their chances of finding a job.

However, there are some of them who were not sure

that computer programming is an easy subject.

DISCUSSION

This study focused on the development of a model which blends learning environment using engineering design to learn computer programming course. For this purpose, the BLE-EDC model has been designed and developed to be used for the course. The model has been formed by combining traditional face-to-face and online learning. The important goals of the BLE-EDC model were to enhance 4 necessary outcomes for learning computer programming which are: programming conceptual knowledge, program analytical skills, problem-solving by programming skills, and attitude in learning programming. The online course aimed to make the students gain the programming conceptual knowledge and some program analytical skills. While face-to-face learning intended to enhance the program analytical skills and problem-solving using programming skills by applying the engineering design concept to the learning activities. In addition, the learning motivation was delivered to students both online and through face-to-face approach to achieve competence in learning programming and drive them to meet the goals of learning.

According to the results obtained from the usage, the post-test scores of programming conceptual, program analytical skills, and problem-solving using programming were significantly increased compared with the pre-test. This shows that the BLE-EDC model can enhance the knowledge and essential skills for learning programming. While comparing between control and experiment group, the experiment group post-test score of program analytical skills and problem-solving by programming were significantly higher than the control group. This revealed that the BLE-EDC model can develop these skills better than a traditional face-to-face learning environment, these results coincide with the findings by other researchers who used the engineering design concept to learning activities (Evans et al., 2016; Dixon and Johnson, 2012; Strobel and Carr, 2011; Robert et al., 2009). However, the programming conceptual knowledge scores of control and experiment groups are not significantly different. This showed that the BLE-EDC model can enhance learners knowledge as well as the traditional face-to-face. The reason that the students' achievement learned by BLE-EDC model was not different with traditional face-to-face learning may come from the less of students' self-regulation in online learning which causing the blended learning is not as successful as should be (Cigdem, 2015). It was also found that the learners in experiment group have a good attitude in learning programming after learning with the developed model. In accordance with Brooks (2009) who concluded that a good blended learning environment affected the positive attitude of learners.

The BLE-EDC model uses a low cost to implement because it uses already existed tools and equipment. According to the Thai government policy B.E.2558 "Reduce instructional time, increase learning time", it allows the students to have more flexible time to study and understand the programming knowledge because the time to learn is not just limit to in the classroom. In addition, the lecturers can have more time to prepare the example codes and problems statement to enhance higher skills of the students. The BLE-EDC model also gives a good attitude in learning programming to the learners which are an important success factor to be a good programmer.

A problem found in this research is that there are some students who did not clearly understand will meet the problem in classroom activities. So the lecturers should announce that if they have any unclear contents, they should ask immediately.

CONCLUSION AND SUGGESTIONS

The findings suggest that the task should be improved as follow. Firstly, the way to motivate the students should be emphasize with showing the life quality of success programmers and the benefits of being a programmer. This is considered to be the best way to build the internal motivation of the students. Secondly, the lecturer should find the example codes from the expert programmers to let the learners absorb good coding styles. Thirdly, to enhance the problem-solving by programming skills, the learner must practice regularly so the lecturer should find a variety of problems for them. Finally, the BLE-EDC model conforms to the Thai government policy B.E.2558 "Reduce instructional time, increase learning time".

Conflict of Interests

The authors have not declared any conflict of interests.

ACKNOWLEDGEMENTS

This research was supported by Faculty of Education, Mahasarakham University and Faculty of Science, Bansomdejchaopraya Rajabhat University.

REFFERENCES

Abbas EZ, Tim L, Nigel B (2009). Blended Teaching and Learning of Computer Programming Skills in Engineering Curricula. Adv. Eng. Educ. 1(3):1-18.

Alducin-Ochoa JM, Vazquez-Martinez AI (2016). Hybrid Learning: An Effective Resource in University Education? Int. Educ. Stud. 9(8):1-13.

Allison I, Orton P, Powell H (2002). A virtual learning environment for introductory programming. Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, Loughborough University, UK. pp. 48-52.

Asunda PA, Hill RB (2007). Critical Features of Engineering Design in Technology Education. J. Ind. Teacher Educ. 44(1):25-48.

Bonk CJ, Graham CR (2005). The handbook of blended learning: Global perspective, local designs. San Francisco, CA: Pfeiffer Publishing.

Boylea T, Bradleya C, Chalkb P, Jonesb R, Pickardc P (2003). Using Blended Learning to Improve Student Success Rates in Learning to Program. J. Educ. Media 28(2-3):165-178.

Cigdem H (2015). How Does Self-Regulation Affect Computer-Programming Achievement in a Blended Context? Contemporary Educ. Technol. 6(1):19-37

Denning PJ, McGettrick A (2005). Recentering Computer Science. *Commun.* ACM 48(11):15-19.

Deperlioglu O, Kose U (2013). The Effectiveness and Experiences of Blended Learning Approaches to Computer Programming Education. Computer Appl. Eng. Educ. 21(2):328-342.

Dixon RA, Johnson SD (2012). The Use of Executive Control Processes in Engineering Design by Engineering Students and Professional Engineers. J. Technol. Educ. 24(1):73-89.

Djenic S, Krneta R, Mitic J (2011). Blended Learning of Programming in the Internet Age. IEEE Transact. Educ. 54(2):247-254.

Dym CL, Agogino AM, Eris O, Frey DD, Leifer LJ (2005). Engineering design thinking, teaching and learning. J. Eng. Educ. 103(1):103-120.

Evans MA, Schnittka C, Jones BD, Brandt CB (2016). Studio STEM: A Model to Enhance Integrative STEM Literacy Through Engineering Design. Contemp. Trends Issues Sci. Educ. 44:107-137.

Hadjerrouit S (2008). Toward a Blended Learning Model for Teaching and Learning Computer Programming: A Case Study. Infor. Educ. 7(2):181-210.

Ismail MN, Ngah NA, Umar IN (2010). Instructional Strategy in the Teaching of Computer Programming: A Need Assessment Analyses. *Turk.* Online J. Educ. Technol. 9(2):125-131.

Jenkins T (2001). The Motivation of Students of Programming. Proceedings of Innovation and Technology in Computer Science Education. Canterbury, UK, 53-56.

Jenkins T (2002). On the Difficulty of Learning to Program. Proceedings of the 3rd Annual HEA Conference for the ICS Learning and Teaching Suport Network. pp. 1-8.

Kwon DY, Yoon IK, Lee WG (2011). Design of Programming Learning Process using Hybrid Programming Environment for Computing Education. KSII Transactions on Internet Infor. Syst. 5(10):1799-1812.

Miliszewska I, Tan G (2007). Befriending Computer Programming: A Proposed Approach to Teaching Introductory Programming. Infor. Sci. Infor. Technol. 4:277-289.

Mohorovicic S, Strcic V (2011). An Overview of Computer Programming Teaching Methods. Proceedings of the 22nd Central European Conference on Information and Intelligent Systems, Faculty of Organization and Informatics, Varazdin. pp. 47-52.

Robert W, Phillip CS, Soo JK (2009). Essential Concepts of Engineering Design Curriculum in Secondary Technology Education. J.f Technol. Educ. 20(2):65-80.

Robins A, Rountree J, Rountree N (2003). Learning and Teaching Programming: A Review and Discussion. Comput. Sci. Educ. 13(2):137-172.

Ryberg T, Dirckinck-Holmfeld L (2010). Analysing digital literacy in action: A case study of a problem-oriented learning process. In R. Sharpe, H. Beetham & S. De Frietas, eds. Rethinking Learning for a Digital Age: How Learners are Shaping their own Experiences. Abingdon: Routledge. pp. 170-183.

Sankas S (2010). Development of a Content and Activity Blended Learning Model for Computer Programming. Unpublished doctoral dissertation, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand.

Sarpong KA, Arthur JK, Owusu PY (2013). Causes of Failure of Students in Computer Programming Courses: The Teacher - Learner Perspective. Int. J. Comput. Appl. 77(12):27-32.

Strobel J, Carr RL (2011). Integrating engineering design challenges into secondary STEM education. 1st Integrated STEM Education Conference (ISEC), 2 April 2011.

Von Bertalanffy L (1968). General System Theory: Foundations, Development, Application. Newyork, NY: George Braziller.