Dr.MGR-Janaki College of Arts & Science for Women
R.A.PURAM, Chennai-600028.
Accredited with 'A' Grade by NAAC, An ISO 9001:2015 Certified Institution
Affiliated to the UNIVERSITY OF MADRAS

Book of Proceedings

International Conference on

SYNERGIZING VISION SYSTEMS *WITH* Analytical Intelligence

Editors Name
Dr. V.R.Viju
Ms. N.S.Kothai

CORAL
PUBLISHERS & DISTRIBUTORS

SYNERGIZING VISION SYSTEMS WITH ANALYTICAL INTELLIGENCE

Editors: Dr. V.R.Viju / Ms. N.S.Kothai

CORAL PUBLISHERS & DISTRIBUTORS

Coral Estate: No. 1-A, Morai Sarath Kandigai Main Road, Morai Pulikuthi Village,
Nagoor Andavar Nagar, Avadi, Chennai - 600 055
E-mail: coralbooks2016@gmail.com Mob: 90430 50666

ISBN : 978-93-47563-55-3

First Edition : 2026

Layout by: Gemini Design 9840487404

Printed by: Mani Offset, Chennai

Contents

BRIDGING THE THEORY-PRACTICE GAP: EVALUATING THE IMPACT OF APPLICATION-BASED TEACHING METHODS ON STUDENT OUTCOMES IN INTRODUCTORY PROGRAMMING

N.S. Kothai,
Research Scholar, Department of M. Sc (Computer Science),
Vels Institute of Science, Technology & Advanced Studies, Chennai

Dr.V. Sumalatha,
Associate Professor, Department of M. Sc (Computer Science),
Vels Institute of Science, Technology & Advanced Studies, Chennai

Abstract

In the new digital classroom era, teachers explain the contextual concepts through traditional standard examples, such as whether a given number is even or odd, the sum of 2 numbers, etc. These explanations do not bridge the gap between context and the real-world applications. Minimizing the gap between the context and real-world problem solving is done by explaining the concepts, such as OTP checking and wrong password attempts. This study has investigated the effectiveness of application-based teaching methods in comparison with the traditional, syntax-centric method in an introductory programming course. A quasi-experimental study was conducted with two groups of 90 first-year computer applications students. Traditional lectures and textbook exercises were taught to a set of 45 control group students, and an application-based curriculum featuring real-world scenarios was taught to the rest of the 45 experimental group students. Application-based teaching methods were found to significantly improve practical coding skills, student engagement, and retention in introductory programming. The findings advocate for a pedagogical shift from a syntax-centric approach to a problem-centric approach in introductory programming.

Keywords: *Introductory Programming, Pedagogy, Student Engagement, Problem-centric, Student-centric, Application-Based Teaching , Retention, Motivation*

1. Introduction

1.1. The Problem Space & Theoretical Background:

The rise of AI, fascinating IoT, and emerging data mining applications in the computer science field create interest among higher secondary school students in pursuing their future as computer science engineers. This will increase the number of students enrolled in the computer science course year by year across the world. Students join the computer science course with a lot of expectations that their fields are closely related to the real world.

Each course has been designed to include all the students who start their academic journey. Among the semesters, the first semester is crucial for the students. If they are exposed to the needs of the course at the onset of the course, it boosts their morale and helps them remain motivated and fine tune their technical skills in the forthcoming years. If they fail to do so, then the outcome will be lacking in quality.

Teaching programming concepts in the introductory programming should aid in minimizing the gap between the context and the real-world applications. If the gap is less, it will create more interest, and it will fulfil their expectations also. This paper takes us through a new application-based learning teaching concepts in introductory programming.

This study examined the outcome of implementation of an Application Based teaching methodology in programming learning, in addition to some findings that were noted while performing the systematic literature review framework.It focused on the challenges encountered in teaching programming as a basis for system development, as well as the advantages and disadvantages related to system characteristics, technology, features, and the effectiveness of the proposed teaching system. This research generated conceptual insights synthesized from relevant literature sources. The findings of the study highlighted whether the Application based teaching systems in programming education could effectively address the difficulties faced during the learning process. This study aims to investigate whether the implementation of the developed teaching systems that have been used for the past years could overcome the problems faced by the students who take such a coding course. This work also examines the published papers on the scarcity, advantages, technology, features, and the final result of the used systems.

The remainder of this paper is organized as follows. The next set of sections describes the procedure used to conduct the literature review, sample strategies used in the proposed teaching system, the findings and evaluation derived from the reviewed studies. The final section sets the conclusion and future work.

1.2. Research Objectives and Questions (ROs/RQs):

- RO1: To design an application-based teaching module for an introductory programming course.
- RO2: To compare the effectiveness of this module against a traditional teaching method.
- RQ1: What is the impact of application-based methods on student motivation and self-efficacy?
- RQ2: How do application-based methods affect students' practical programming skills compared to traditional methods?

The whole paper significantly proves that this Application based teaching approach will improve both the retention capacity and the practical coding skill of the first year students.

2. Literature Review

This literature review was carried out following the Systematic Literature Review (SLR) framework [12]. The review was done systematically over articles in journals, proceedings, and book chapters that were published between 2001 and 2025. The papers were selected mainly by using the keywords and hints that were relevant and could achieve the purpose of the study. The keywords used were "introductory programming course", "teaching and learning", "Enhancing coding skills"," real world learning"," teaching programming" and "programming skills". After searching the literature using several pre-determined keywords, twenty-four papers were obtained from several sources.

In the Internet era, students have increasingly lost interest in traditional lectures; consequently, their learning motivation and exam performance have decreased. The widespread adoption of learner-centered teaching methods that address this issue faces certain barriers, including: 1) the significant faculty effort necessary to prepare e-learning materials; 2) significant extra time required for active online communication with students; 3) student resistance to

taking an active role in their education; and 4) lecturers' common belief that learner-centered teaching activities do not allow discussion of all the required topics[1]

Its abstract nature makes it difficult to teach, learn, understand and assimilate. Moreover, most introductory programming (IP) modules are now offered by several students with diverse backgrounds (computing and non- computing) in most institutions as a means of ensuring digital core competencies in all disciplines. This has led most of these students to face several challenges leading to a lack of motivation, significant failure and dropout rates despite several teaching and learning methods and tools that have been proposed, designed and developed [2].

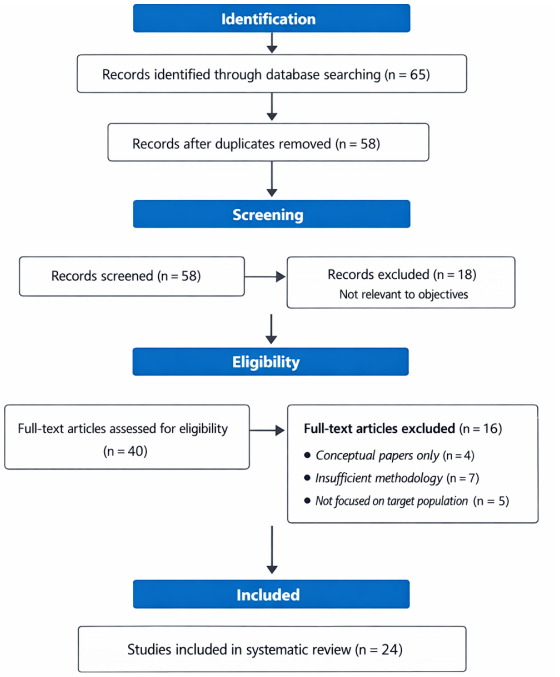

Figure 1: PRISMA Flow-Diagram

Students exhibit various misconceptions and other difficulties in syntactic knowledge, conceptual knowledge, and strategic knowledge. These difficulties experienced by students are related to many factors including unfamiliarity of syntax, natural language, math knowledge, inaccurate mental models, lack of strategies, programming environments, and teachers' knowledge and instruction. However, many sources of students' difficulties have connections with students' prior knowledge. To better understand and address students' misconceptions and other difficulties, various instructional approaches and tools have been developed.[3]

We stated that programming is difficult. Several researchers all over the world confirm this statement. This problem has stimulated researchers to find ways to help students learn and teachers to teach programming. Our experience in teaching introductory programming and with students with very particular characteristics tells us that it is important to change the way we teach in such a competitive and constantly evolving area. We also want to leave our small contribution in the area with the presentation of our idea.[4]

Creating innovative teaching methods and shifting the lecturer-centered learning to students-centered will enhance the students' skills in programming. Exposing the students to the real problems and proposing the solutions by applying related theories could lead to higher competencies among students, generate a better quality of future programmers and fulfil industrial demands. [5]

Problem-Based Learning (PBL) employs methods designed to build practical problem-solving skills. The core principle is simple: to enhance programming competency, students must be confronted with real-world challenges and tasked with designing and proposing effective solutions. [8]

Test driven teaching is a Learning - oriented and learner-directed learning process. The author recommended it as the best method for introductory programming courses. Weekly 5 hour of class that included supervised learning and assignment review and an expectation that students would spend 5 hour on independent study between classes. In that assignment students attempted to solve the exercise based on the concerns raised by the students.[6].

Major Weakness of the students in the introductory programming are phases of programming stage, problem solving skills, ineffective pedagogy and persona traits and

attitude. Development of effective teaching and learning material is needed to overcome from the weakness.[7]

Numerous research activities in studying the difficulties in Teaching and Learning introductory programming have been carried out by many researchers in the past 2 decades. The substantiation of this situation is well documented [9].

Exposing the students to real problems and proposing the solutions by applying related theories could lead to higher competencies among students, generate a better quality of future programmers and fulfil industrial demands.[10]

We need to motivate students, engage them in the process, and induce them to display an interest in learning to be effective programmers.[11]

Taking programming courses requires some skills, so at the end of the courses, the student could develop a product or more that could be beneficial for others. If they could, the product could give a big prospect in the future; it could be some newly developed systems or some innovation that could help people. It requires skills in critical thinking, problem-solving, computational thinking, and new system designs, as is said in research. There have been some related resources that discuss programming learning courses. However, none of those sources said that the learner encounters difficulties understanding the study that requires an abstract concept, such as programming [13, 14,15]. Students often faced some difficulties in making a program algorithm or syntax which resulted in an error result [13].

Strong programming skills can enable individuals to navigate and understand the digital environment more effectively. Thus, individuals can understand how these technologies work and how they can be used and manipulated. Computer programming is important for problem-solving and critical thinking [16][17]. In practical teaching increased a series of simple system design based on project-driven, and each person as a group, so that every student can really participate in and get the exercise, greatly enhanced their confidence [18], including the use of online systems that provide learners with additional practice opportunities to gain greater experience in developing computer programs. Online-based educational systems have been shown to be as effective as, or even more effective than, traditional teaching methods [19]. As concluded in [20], increasing the number of practical lessons positively influences the development of students' abilities in programming and numerical courses by allowing them to gain deeper experience in understanding key concepts. Furthermore, greater engagement in practical activities, learning situations, and instructional materials leads to improved learning outcomes [21]. Various teaching approaches are used to provide effective programming education to learners. Instructional approaches such as hands-on coding, project-based learning, pair programming, problem-based learning, and game-based learning are among the current approaches used in programming education in recent years. [22[23[24].

Table 1: Overview of Findings

Findings	References
The abstract nature of Introductory Programming(IP) makes it difficult to teach, learn, understand and assimilate.	[1],[4],[13],[14],15]
Students gradually lost interest in traditional lectures	[1],[4]
Significant failure and dropout rates have been noticed in Introductory Programming(IP).	[1],[2]

Students exhibit various misconceptions and other difficulties in syntactic knowledge, conceptual knowledge, and strategic knowledge.	[3]
Motivation is highly needed in learning programming	[1],[2],[11]
Significant faculty effort is necessary, Development of Effective teaching and learning material is needed.	[1],[5],[7]
Usage of Technology and Practical approach will leads increase the programming proficiency	[16],[17],[18],[20]
Practical based assignments, greater engagement in practical activities, learning situations, and instructional materials leads to improved learning outcomes	[6],[21]
Different teaching methodologies have been proposed other than traditional teaching methodologies	[19],[22],[23],[24]
Expose to Real world problem lead to higher competencies, generate a better quality of future programmers and fulfil industrial demands	[5],[8],[10]

3. Methodology

Minimum of 60 practical and 60 theory hours has been prescribed by the University of Madras for Introductory programming. Teachers should cover the entire syllabus within the stipulated time and need to allocate the time for evaluation and remedial program also. Teachers can start practice to explain the programming concepts through real world applications that will motivate them and enhance the students' programming skills.

We have selected 90 female students from 1st year Computer Applications with different backgrounds such as computer science/, Mathematics/ other background, with/without prior knowledge on programming for this research work. We have selected Python as an introductory programming language. Initially we have split them into 2 groups as control groups (I BCA A) and as an experiment group (I BCA B). Duration for these 60 theories + 60 practical sessions is from July 15th to October 6th.Traditional Teaching methods have been adopted for the control group. Application based teaching was introduced to the experiment group. Within 3 months 2 monthly exams and final semester exams were conducted for both the groups. At the end of the course a Google form based on Application based Teaching 12 questions were framed and circulated to both groups. Sample concepts that have been defined here will increase the understandability of the concept very well.

Concept 1: Computer Hardware

The teacher should draw a neat block diagram of the computer on the board. Teachers can start to explain the usage and functional work of each part of the block diagram of the computer, ask the students to collect the laptop/computer sales advertisement, and relate each and every part with the block diagram of the computer. Compare and contrast the various configurations of the computer and choose the best one among them.

Topic	Traditional Approach	Application Based Teaching
Computer Hardware	Block diagram of the computer	One week assignment on the Advertisement of the sales of Laptop/ Computer. 3 hours Lab Visit. Student start explore to the physical parts of the computer

Concept 2: Computational Problem Solving

First teachers should explain to students how the problem should be identified from its own environment and then how we must give solutions through computer programming. Give one week Self pace assignment on Apps on the Phone.

Each student should select one App from the smart phone and submit a page report. The report should give answers for the following questions 1. App Name with creator details 2. Problem identified before creating the App 3. Users. 4. How they fulfill it 5. Suggestions for some future enhancement in the App.

Topic	Traditional Approach	Application Based Teaching
Computational Problem solving	Traveling salesman problem, river crossing Plan, Chess.	One week Case study on Apps/ websites/ software's.

Concept 3: Conditional statements

Teachers usually explain the conditions statement through the standard example ike whether the given number is positive or negative, even or odd. In this Application based Teaching, teachers can start to explain the same concepts through some real word examples such as OTP, 2SV verification, Login Credentials etc..

Topic	Traditional Approach	Application Based Teaching

Simple If	Task: Positive or Negative
	Read a
	if(a>0):
	print("A is positive Number")
	else:
	print("A is a Negative Number")
		Task 1: OTP Verification
		Read rcvd_otp
		If (rcvd_otp == OTP):
		Open the App
		else:
		Display a warning message
		Task 2: PIN Number Checking
		Task 3: QR Code Scanning
		Task 4: Finger print recognition

Topic	Traditional Approach	Application Based Teaching

Nested If	Greatest among three numbers
	Read a,b,c
	if(a>b):
	if(a>c):
	print("A is Largest")
	else:
	print("C is largest")
	else:
	if(b>c):
	print("B is Largest")
	else:
	print("C is largest")
		Login Page
		Task 1
		Read User_name and Pass_word
		If (User_name == UserName):
		if(Pass_word==password):
		print("Login Successfull")
		else:
		Display wrong password
		else:
		Display wrong user name
		Task 2:
		2 Step Verification
		Task 3:
		E-Mail Login

Concept 4: Looping statements

Usually teachers explain the conditions statement through the standard example whether the given number is positive or negative, even or odd in 2SV verification entering the accurate OTP is important

Topic	Traditional Approach	Application Based Teaching
For Loop	Print First n Number Read n For i in range(n): print(i)	Task 1: Waiting time calculations For i in range(45): print((45-i),"seconds remaining") Task 2: File content Transfer Task 3: Waiting time calculation Task 4: OTP recognizing time.

Topic	Traditional Approach	Application Based Teaching

While	Print First n Number
	Read n
	i=1
	while(i<n):
	print(i)
	i+=1
		Task 1:
		Count Password chances
		Read pass_word
		i=1
		while (pass_word!=password):
		If (i==3):
		print("Account has been locked ")
		print((3-i),"chances are there")
		i+=1

Concept 5: Functions/ sub procedure

Theoretically functions are taught as a logical division of our main program. So logically divide the program into multiple parts, assign user defined names for that. The main advantage is that its easy for debugging, repeated and convenient calls.

Topic	Traditional Approach	Application Based Teaching
Function	Def sum(): Statements; Return Def sub() Statements Return Main function	ATM Screen Def withdrawal(): Statements; Return Def deposit(): Statements; Return Def mini_statement(): Statements; Return

According to this Application based approach we can take ATM Screen as an example. the label present in the ATM screen such as withdrawal, deposit, mini statement, balance enquiry, pin setting. Each label will be associated with functions

Concept 6: Exception Handling

Theoretically we give definition for exception handling as An event that occurs during the execution of a program that disrupts the normal flow of instructions and standard example for this is divide by zero.

Topic	Traditional Approach	Application Based Teaching
Exception Handling	Divide by Zero Array Index out of bound	404 web resource is not available Web site Under construction

Concept 7: Modules

A module is a file containing a set of code, such as functions, classes, and variables, that can be reused in other programs

Topic	Traditional Approach	Application Based Teaching
Modules	Packages such as sum & multiply	We can keep the same Login Screen, payment gateway for all our projects

4. Results & Discussions

This analysis treats the student responses as a dataset to identify patterns and correlations. The questions about programming concepts (If, For loops, Functions, etc.) are proxies for computational thinking and logical structuring, which are the bedrock of ML.

Key Observations:

- **Depth and Variety of Examples**: CS background students consistently provided more varied and specific examples.
- **CS Students:** Often mentioned advanced concepts like "Tower of Hanoi" (recursion), "Pattern checking" (nested loops), "2SV checking," and "Fibanocci Series." Their answers for "Functions" and "Modules" were more application-oriented (e.g., "ATM Options," "Menus in Word"). Whereas Non-CS Students: Answers were more fundamental. For "If statement," they primarily listed "check even or odd." For "Functions" and "Recursion," "Factorial of a number" was the most common, almost exclusive answer. They showed a higher frequency of "None of the above" for modules and recursion.
- A CS background provides a significant head start in ML. These students show a stronger grasp of the algorithmic decomposition and systematic problem-solving required to frame a problem for an ML model, prepare data (which involves loops and conditionals), and structure the resulting codebase.
- Affinity for Algorithmic Problems: Students with a Mathematical background showed a clear preference and awareness for classic algorithmic problems.
- Mathematical Background: Dominated the selection of "Tower of Hanoi" for recursion, a problem deeply rooted in mathematical logic and sequence. They also more frequently mentioned "Multiplication Table printing" and "Greatest among 3 numbers."
- Non-Mathematical Background: Showed a strong tendency towards "Factorial of a given number" as their primary example for both functions and recursion, indicating a grasp of the concept but potentially a more limited repertoire of examples.
- A Mathematical background is a strong predictor of comfort with the core of ML. Understanding models like regression, neural networks, and evaluation metrics requires comfort with functions, algorithms, and statistical thinking. Students without this background may find it easier to grasp the application and deployment of ML (the "what") but could struggle with the underlying principles (the "why").

Control Vs Experimental group

This dimension looks for patterns specific to the class section, which could reflect differences in teaching methodology, peer group, or overall classroom culture.

Key Observations:

- Prevalence of "Tower of Hanoi": This is the most striking difference.
  - Experiment Group: The vast majority of students in BCA B selected "Tower of Hanoi" for the recursive functions question.
  - Control Group: Responses in BCA A were split between "Tower of Hanoi" and "Factorial of the given number." This suggests that recursion was taught or emphasized differently between the two sections, with BCA B focusing more on a classic, complex example.
- Diversity in "For" and "While" Loop Applications: BCA A students showed a slightly wider range of examples for loops, including "Multiplication Table printing" and "Pattern checking." BCA B students heavily consolidated around "Traffic Signal Waiting Time" and "No of times left - Warning Message." This could indicate different teaching examples used by the instructors.
- Favorite App Category: While both classes listed popular apps (WhatsApp, GPay, Uber), BCA B had more unique entries like "Blink It," "Om Tamil Calendar," "DOC Scanner," and "Truecaller," suggesting a slightly more diverse range of interests or exposure.

Students from all backgrounds show an understanding that programming logic is embedded in daily applications, which is a positive first step toward grasping applied AI/ML

6. Future Work

Most of the students think that programming is a difficult concept till their pre-final semester, if they fail to learn it properly. Once they start work on the project in their final semester the programming skill feels easier for them. Creating interest in programming will enhance their programming skills. One of the ways to create interest is to relate the basic concepts of the programs to the real-world application.

Application-based teaching methods, which emphasize real-world problem-solving and tangible outcomes from the outset, are a powerful strategy for improving student success in the challenging domain of introductory programming. By making code meaningful, we can foster not only better programmers but also more motivated and persistent learners.

This research paper covers only some basic concepts in programming. Still, we can implement Application based teaching in advanced concepts in programming. This paper has suggested assignments for individuals. Future researchers can implement the same concepts for different learning styles.

REFERENCES

[1] Debiec, Piotr. (2017). Effective Learner-Centered Approach for Teaching an Introductory Digital Systems Course. IEEE Transactions on Education. PP. 1-8. 10.1109/TE.2017.2729498.

[2] A.K. Mbiada, B.Isong, F.Lugayizi and A.Abu-Mahfouz, "Introductory Computer Programming Teaching and Learning Approaches: Review," 2022 International Conference

on Electrical, Computer and Energy Technologies (ICECET), Prague, Czech Republic, 2022, pp. 1-8, doi: 10.1109 /ICECET55527.2022.9873427. keywords: {Learning systems; Education; Programming; P networks; Faces; Computer programming;Teaching; and learning; Students; IP module},

[3] Qian, Yizhou & Lehman, James. (2017). Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. ACM Transactions on Computing Education. 18. 1-24. 10.1145/3077618.

[4] Figueiredo, José & García-Peñalvo, Francisco. (2018). Building Skills in Introductory Programming. 46-50. 10.1145/3284179.3284190.

[5] Kadar, Rozita & Wahab, Naemah & Othman, Jamal & Shamsuddin, Maisurah & Mahlan, Siti. (2021). A Study of Difficulties in Teaching and Learning Programming: A Systematic Literature Review. International Journal of Academic Research in Progressive Education and Development. 10. 10.6007/IJARPED/v10-i3/11100.

[6] Tirronen, V., & Isomottonen, V. (2011). Making teaching of programming learning-oriented and learnerdirected (pp. 60–65). Proceedings of the Koli calling, ACM: Koli.

[7] Cheah, Chin-Soon. (2020). Factors contributing to the difficulties in teaching and learning of computer programming a literature review. Contemporary Educational Technology. 12. ep272. 10.30935/cedtech/8247.

[8] Medeiros, Rodrigo & Ramalho, Geber & Pontual Falcão, Taciana. (2018). A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. IEEE Transactions on Education. PP. 1-14. 10.1109/TE.2018.2864133.

[9] Bennedsen, J. 2008. Teaching and learning introductory programming : a model-based approach. University of Oslo.

[10] Kadar, R., Wahab, N. A., Othman, J., Shamsuddin, M., & Mahlan, S. B. (2021). A Study of Difficulties in Teaching and Learning Programming: A Systematic Literature Review. International Journal of Academic Research in Progressive Education and Development, 10(3), 591–605.

[11] Anthony Robins , Janet Rountree & Nathan Rountree (2003) Learning and Teaching Programming: A Review and Discussion, Computer Science Education, 13:2,137-172

[12] Okoli C. A Guide to Conducting a Systematic Literature Review of Information Systems Research. Communications of the Association for Information Systems. 2015;37(1):879–910.

[13] Yuana RA, Leonardo IA, Budiyanto CW. Remote Interpreter API Model for Supporting Computer Programming Adaptive Learning. Telkomnika. 2019;17(1):101–8.

[14] Lopez AR, MuñozDF. Increasing Practical Lessons and Inclusion of Applied Examples to Motivate University Students during Programming Courses. Procedia- Social and Behavioural Sciences. 2015;176(1):552–64.

[15] Lahtinen E, Alamutka K. A Study of the Difficulties of Novice Programmers. In: Proceedingsofthe10thAnnualSIGCSEConference on Innovation and Technology in Computer Science Education; 2005 Jun 27-29; Monte de Caparica, Portugal. Association for Computing Machinery; 2005. p. 14–8.

[16] R. Mathew, S.I. Malik, R.M. Tawafak, Teaching problem solving skills using an educational game in a computer programming course,Informatics in Education, 18 (2) (2019), pp. 359-373

[17] X.M. Wang, G.J. Hwang, Z.Y. Liang, H.Y. Wang,Enhancing students' computer programming performances, critical thinking awareness and attitudes towards programming: An online peer-assessment attempt Journal of Educational Technology & Society, 20 (4) (2017), pp. 58-68

[18] Cui L. Research on Educational Reform of Java Programming. In: Proceedings of the 6th International Conference on Mechatronics, Computer and Education Informationization; 2016 Nov 11-13; Shenyang, China. Atlantis Press; 2016, p. 710–3

[19] Weber G, Brusilovsky P. ELM-ART: An Adaptive Versatile System for Web-based Instruction. International Journal of Artificial Intelligence in Education. 2001;12(1):351–84.

[20] Lopez AR, MuñozDF. Increasing Practical Lessons and Inclusion of Applied Examples to Motivate University Students during Programming Courses. Procedia- Social and Behavioural Sciences. 2015;176(1):552–64

[21] Lahtinen E, Alamutka K. A Study of the Difficulties of Novice Programmers. In: Proceedings of the 10thAnnualSIGCSEConference on Innovation and Technology in Computer Science Education; 2005 Jun 27-29; Monte de Caparica, Portugal. Association for Computing Machinery; 2005. p. 14–8

[22] J.C. López-Pimentel, A. Medina-Santiago, M. Alcaraz-Rivera, C. Del-Valle-SotoSustainable project-based learning methodology adaptable to technological advances for web programming Sustainability, 13 (15) (2021), p. 8482

[23] S. Malik, M. Al-Emran, R. Mathew, R. Tawafak, G. AlFarsi, Comparison of E-learning, M-learning and game-based learning in programming education–a gendered analysis, International Journal of Emerging Technologies in Learning (IJET), 15 (15) (2020), pp. 133-146

[24] A. Sullivan, A. Strawhacker, Screen-free STEAM: Low-cost and hands-on approaches to teaching coding and engineering to young children STEAM in early childhood education and care, Springer International Publishing, Cham (2021), pp. 87-113