Dealing with Coding Challenges Through Digital Platforms: Assessing Their Effectiveness in Skill Development

Dr.Mohammed Yahya Alghamdi ,Department of Computer Science, Faculty of Computing & Information, Al-Baha University, Al-Aqiq, 65779-7738, Saudi Arabia

myahya@bu.edu.sa

ABSTRACT

This research paper investigates the challenges faced by diploma-level IT students at Al-Baha University in understanding programming concepts, writing code, and detecting errors. It also examines the effectiveness of e-learning platform W3Schools in enhancing students' programming skills. Through a comparative analysis of students using e-learning resources versus those employing traditional learning methods, the study demonstrates the impact of the digital approach in addressing common programming challenges. Both pre- and post-experiment surveys showed positive gains in the Interactive Group, and the statistical data also confirmed a high level of significance for all the programming skills measured. The significance of this paper lies in how the research demonstrates that e-learning can reduce the barriers faced by programming students and contribute to the literature on learning techniques in the context of today's technological landscape. Consequently, the results provide a compelling argument for the necessity of incorporating interactive e-learning resources into the programming curricula to improve student performance and enhance confidence in their coding skills. The findings of this study offer valuable insights into the use of digital approaches in education and suggest a potential way to overcome common learning challenges on a large scale. This work aims to contribute to the ongoing discussion on the impact and effectiveness of digital practices in education. Future research recommendations focus on assessing the long-term value of e-learning in enhancing the learning experience for programming students and better preparing them for future technologies.

Keywords: Programming Skills; W3Schools; Problem-Solving; Programming Challenges, Software Development.

1. INTRODUCTION

This is the age of computers where we are direly in need of computer science, especially programming language to be more equipped to better deal with situations in this technology-controlled world. However, there are several challenges that a computer science student can come across in grasping issues to do with programming language and the like. Software development is the systematic use of scientific principles to design, develop, implement, and maintain software to execute stated tasks using languages such as Python or Java. Professional programming skills is cited as one of the most valuable skills that need to be developed in the world that is turn around with technology and information. It needs in the process of formation of mental and logical skills of students and give them a possibility to solve problems in rather creative ways. Nonetheless, learning programming involves many problems and struggles which must be acknowledged and solved. As a result, current research attempts to examine these challenges and determine how to

overcome them. In this research, the programming learning experience of the students will be assessed along with the challenges faced while doing so. In Computer Science subjects, the difficulty levels differ based on the teaching and learning processes [1]. The difficulty level for the topic of Computer Science is determined by the participants' backgrounds and experiences. Computer Science lecturers must strike a balance between theory and practical applications. It is also necessary to provide students with opportunities to implement their knowledge and skills by providing problem-solving exercises by practical applications on well-defined scenarios [2]. Problem-solving exercises, tasks designed to enhance critical thinking and the ability to solve complex issues as well as coding project assignments involving the creation of software applications by writing and implementing code are essential for mastering programming concepts. Practical tasks that involve working with programming languages or software development tools simplify and elevate the learning process. Computer Science courses may appear as interesting yet challenging to pursue [3], [4]. It requires problem-solving abilities, logical thinking, as well as a willingness to continuously learn and adjust to new technologies. Learning intricate ideas and algorithms requires perseverance and practice. For instance, programming poses a common problem in the Computer Science curriculum. Algorithms are step-by-step procedures for solving problems efficiently, crucial for developing optimized code in various applications. It is common for students to fail and drop out, especially during the early stages [5], [6]. A variety of tools should be made accessible to support Computer Science students especially to practice programming. From student perspective in their academics, a large fraction of students is lacking problem solving skills [7],[8],[9], [10]. Programming is difficult as it requires time and effort, focused knowledge, and skills. Acquiring these skills is an arduous trial-and-error process and requires persistence [11]. Hence, a series of skills, as well as persistence, is essential on the journey to becoming a proficient programmer. Understanding programming syntax is just the initial step in the challenging journey of developing effective programs. Students must delve deeper into understanding the syntax and the logic of computer programming to address complex real-world problems [7]. Conventional teaching materials such as printed books prove ineffective for grasping the dynamic nature of Computer Science subjects such as programming [12],[13]. Regarding pedagogy, it is recommended to start with simple complex programming [8]. Students' negative perceptions influence their learning attitudes toward computer programming, undermining their intrinsic motivation to pursue success [14],[15], [9]. This research seeks to reveal the programming challenges faced by diploma-level IT students at Al-Baha University as they learn to code and understand programming concepts. It also examines how effectively W3Schools addresses the programming issues encountered by students. Furthermore, this study evaluates how W3Schools helps students learn, comparing their outcomes with those of traditional educational methods before and after implementing the digital approach. The research contributes to the development of future learning methods and enhances programming education by analyzing the impact of e-learning on student results using quantitative data.

2. Literature Review

A vast body of literature on computer science education reveals the diverse terrain of obstacles experienced by students, specifically those related to programming, as well as provides metrics on approaches and resources to address these challenges. Computer Science equips students with the

necessary skills for the digital age and has become an essential component of contemporary education [16], [17]. Nonetheless, this area poses distinct difficulties for both students and lecturers. One of the goals of this research study is to shed light on the important areas that require attention and improvement. This review shall examine the challenges associated with teaching and learning Computer Science. One of the many challenges for Computer Science students is that they have to understand abstract concepts and advanced algorithms. A solid foundation and understanding are needed such as algorithms, data structures, and programming paradigms [18]. An example of an optimum teaching technique is to dissect challenging subjects into smaller parts while providing real-life applications for better understanding. The current challenge for both the lecturers and students is the rapid technological development related to Computer Science. Therefore, they need to be kept up to date with the latest developments [19]. There are also challenges in being up to date with the current frameworks, tools, and programming languages [20]. Instead of focusing solely on teaching specific technologies, lecturers should consider focusing on fundamental concepts and problem-solving approaches. Such challenges can subsequently be addressed by providing tools for students to be able to continuously improve their skills by themselves. Another challenge is the common perception that Computer Science is focused on theory and abstract fields with little practical applications. This may have an impact on the students' motivations. For the purpose of highlighting the practicality and the applicability with respect to Computer Science principles, it is proposed that lecturers integrate real-world scenarios as well as hands-on projects into their lesson plans. Supporting the students in coding competitions, engaging in practical projects, or cooperating with industries on actual projects may lead to improvement in the student's enthusiasm and comprehension. Students' enrollment also differs in their gender preference, where there is a thin representation of females in Computer Science [17, 18]. Insufficient infrastructure and resources are a challenging factor in teaching and learning Computer Science. It is vital for educational institutions and universities to invest in software licenses, modern computer labs, as well as additional resources. Furthermore, it is essential to provide lecturers with opportunities for professional development for the purpose of enhancing their Computer Science pedagogical abilities as well as knowledge. According to past research studies that highlight the challenges faced by Computer Science students in a variety of subjects, students find it difficult to learn some aspects of Computer Science, such as programming. Based on the insights gleaned from prior researchers, it has been observed that challenges arise in the initial stages when learning programming. Students encounter challenges in grasping abstract programming concepts such as control structures as well as formulating algorithms to address concrete problems [8], [5], [9]. A lack of solidification regarding abilities, as well as problem-solving skills and logical reasoning, exacerbates these issues during the early phases of learning [7], [10]. As a result, there is a need to shift the focus towards higher-level knowledge, for instance, programming strategies and conditional approaches. It is imperative to introduce metacognitive skills, encompassing knowledge about 'when' and 'why' during the early stages of teaching programming [19]. Several studies have underscored the significance of metacognitive skills pertaining to computer programming courses, identifying them among the factors contributing to students' struggles with programming problems [20], [21], [9]. Conventional teaching methods that rely on static materials, for instance, books, notes, as well as slides, have proven to be ineffective methods of teaching Computer Programming [12]. The inadequacy of static teaching

materials lies in their lack of personalization to cater to specific groups of students, as such materials are developed for a broad, global audience. These materials fail to provide live interactions and dynamic elements to elucidate programming concepts. The ideal scenario is to have an instructor available for immediate feedback and detailed explanations when needed by students. Nevertheless, this is often impractical due to time constraints, limited staff, and large class sizes. The use of static printed materials to teach programming is a hindrance to explaining dynamic software design conceptions, as some students struggle to grasp the dynamic nature of programs due to the inflexibility of static materials [8]. Such a notion is supported by prior research that determined that certain students may prefer solitary learning while some may favor an interactive learning setting [13]. Apart from that, the dynamic learning environment involves activities such as group discussions and peer interactions. Consequently, the current teaching approach that employs static materials does not cater to the various learning styles of students. Therefore, it is imperative for instructors to ensure that their teaching methods can accommodate diverse student groups. In terms of pedagogy, instructors tend to prioritize teaching syntax when it comes to programming languages over promoting problem-solving techniques. The selected programming language used for the course is often based on popularity rather than the suitability of the teaching methods [8]. Opting for an irrelevant programming language just to fulfill educational purposes not only hampers the usefulness of teaching but also heightens the challenges faced by students in grasping the subject [10]. Programming necessitates a high level of analysis and abstraction in thinking to generate effective solutions. This involves implementing specific programming concepts as well as algorithms. Consequently, selecting programming languages based on industry popularity is inappropriate since these languages are primarily designed for professional and specific use instead of supporting educational goals. Computer Science subjects such as programming require persistence, continuous learning, and knowledge from other subjects from time to time. Students often seek solutions or give up when faced with obstacles. Results from previous experiments have demonstrated the existence of a relationship that exists between mathematical problem-solving competencies as well as programming abilities in introductory programming courses [22]. From a psychological perspective, students tend to hold negative perceptions about programming. Such negativity arises from comments, opinions, as well as suggestions from their friends who have previously studied this subject. When students harbor negative thoughts and incorrect impressions, they are more likely to believe that programming is challenging. Consequently, this negatively affects their intrinsic motivation and diminishes their drive to learn [14]. Apart from that, the intensity of these negative perceptions does not only impact intrinsic motivation but also influences students' attitudes toward computer programming. Individual attitudes are shaped by behaviors and indicate the cognitive actions that determine an individual's success or failure in accomplishing a particular task. The difficulties encountered in learning computer programming can be attributed to various factors. Initially, students face hurdles in creating programs to address specific tasks because they struggle with the delineation of distinct functionalities into procedures. Students encounter difficulties in grasping the syntax of programming languages, which is challenging in terms of recalling and comprehending the knowledge [7][15]. Simultaneously, the arduous and time-consuming process of debugging programs further diminishes students' motivation to excel in programming languages. There are also some other challenges in computer programming that are discussed here. The first challenge

is the lack of problem-analysis skills and an understanding of programming concepts. This stems from the fact that students lack essential prerequisites, for instance, discrete mathematics as well as a course in logic programming [23]. The second contributing factor arises from the ineffective utilization of presentation techniques related to problem-solving. Traditional methods such as pseudo code and flowcharts are appropriate in teaching structured programming but fall short when it comes to instructing object-oriented programming, which is the prevalent approach in contemporary programming languages. To address this issue, instructional methods that offer enhanced visualization and explanation are necessary to help students form a mental representation of the problem [9]. The third challenge results from the unsuccessful practice regarding teaching approaches concerning coding practices. Normal teaching approaches are no longer applicable for instructing object-oriented programming, and instructors concur that implementing those distinctive perceptive approaches is essential. Teaching materials that support spatial and visualization abilities are required to aid students in comprehending the processes regarding data as well as control flow. The absence of active student engagement and participation during practical sessions exacerbates the problem, leading to students struggling to grasp computer programming during the learning process. Therefore, the last element for challenges in programming learning relates to the student's understanding and mastering of programming syntax [7], [15]. This is made worse when the students are unable to code programming constructs well. It can be elaborated by the factors of understanding programming concepts, lack of knowledge in syntax, and inability to write efficient programming code [9]. Computer Science is difficult to teach and learn for many reasons, including but not limited to the abstract nature of the subject, the rapid development of technology, the inadequacy in practical application, the gender and diversity gap, and a lack of resources. By implementing practical projects, effective teaching strategies, promoting diversity, as well as infrastructure and resource investments, students studying Computer Science in particular can benefit from a more stimulating and diverse learning environment. Accessing online resources for Computer Science learning supports students and helps them overcome their difficulties in Computer Science subjects. One of the resources to learn Computer Science is Stack Overflow, a well-known online community for programmers, offering an extensive archive of many programming-related issues and knowledge, providing solutions for the international programming community through collaborative learning [24]. Coursera offers a wide range of high-quality computer science courses and degree programs that help students gain a more comprehensive understanding of the subject and practical skills with multiple learning pathways [25]. A large variety of user-created courses designed for practical, hands-on learning experiences designed for diverse skill levels can be found on Udemy [26]. Codecademy is a site that specializes in interactive coding tutorials that are designed to make learning how to code fun and accessible to beginners as well as professional developers, ranging from a gamified learning experience and variety of programming languages and topics covered [27], [28]. Also, it offers interactive coding exercises, including lessons in programming languages like Python, JavaScript, HTML, as well as CSS. It provides a practical learning environment with instant feedback. There are various Computer Science courses available at Khan Academy that are beyond algorithms, data structures, and computer programming [29]. The platform offers quizzes, practice tasks, and video tutorials. In addition, Coursera also offers online courses from top institutions and universities worldwide [30]. There are a variety of subjects taught by experts in the field, such as web

development, machine learning, including artificial intelligence [31]. Another website that offers a choice of selection of Computer Science courses from top universities is edX [32]. It provides activities and quizzes in interactive programming assignments, tests and video lectures. Students can work together to write code and share their works on coding projects through GitHub which is an online hosting service use by computer sciences [33]. W3Schools is an online educational platform that provides resources for students to learn programming through hands-on practice [34]. Consequently, this research examines whether W3Schools can help students address their programming challenges at Al-Baha University. It investigates how the e-learning platform W3Schools supports diploma-level IT students at Al-Baha University in developing programming skills while enhancing their problem-solving abilities and knowledge acquisition. The study validates existing research on how technology improves learning experiences and resolves difficulties through educational programming content.

3. Research Method

This research study used a quantitative research approach to investigate the barriers that diploma-level IT students at Al-Baha University face in learning programming, writing code, and recognizing mistakes. In addition, the study aimed at evaluating the efficacy of the e-learning resources in enhancing the students' performance experimental group compared to the control group during the pre and post treatment periods with the traditional facilitators. The overall goal was to carry out quantitative research based on quantitative data, and then quantify the identified problems to shed a light on those difficulties which the aforesaid students can encounter when learning computer programming, and to compare how these challenges were met in the two groups. In this study, the participants were 100 Diploma level IT students who have already developed adequate programming ability imperative for ascertaining the effects of e-learning and were recruited consecutively. Specifically, out of the total number of students, 50 were assigned to the experimental group, while the remaining 50 formed the control group. Experimental group had 50 students enrolled to e-learning for their programming resources from W3Schools and control group had 50 students following traditional methods. As shown in Figure1, out of the 50 students in the experimental group 32 were female and 18 were male along with the control group, 30 of the 50 students were female while 20 were male. Age distribution of the experimental group is shown in Figure 2 in which 44 students of the total experimental group belongs to the age group of 18-20 years and six students belongs to the age group of 21-23 years. The students in the control group comprised 37 students aged 18 to 20 years and 13 students aged 21 to 23 years. Participants were first informed of the goals of the study, making them aware that the study sought to establish common difficulties in learning programming. Surveys conducted before and after the experiment were utilized to collect data on student difficulties associated with programming. The surveys included questions, detailed in Table 1, designed to assess the difficulties faced by students in programming and problem-solving, specifically focusing on four key challenges: understanding basic concepts, writing programs, learning code, and error recognition. Descriptive statistics were utilized to highlight the major issues faced by students, offering insights into the efficacy of e-learning platforms, such as W3Schools, in alleviating programming difficulties and improving the

educational experience. The investigation sought to find optimal strategies for enhancing student learning in digital environments.

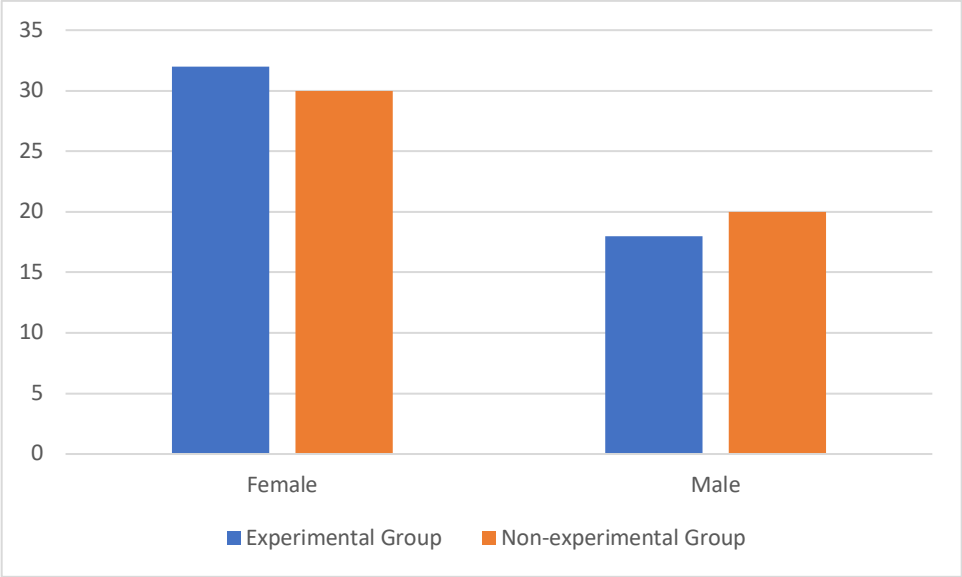

Figure 1 Gender Distribution of Experimental and Control Groups

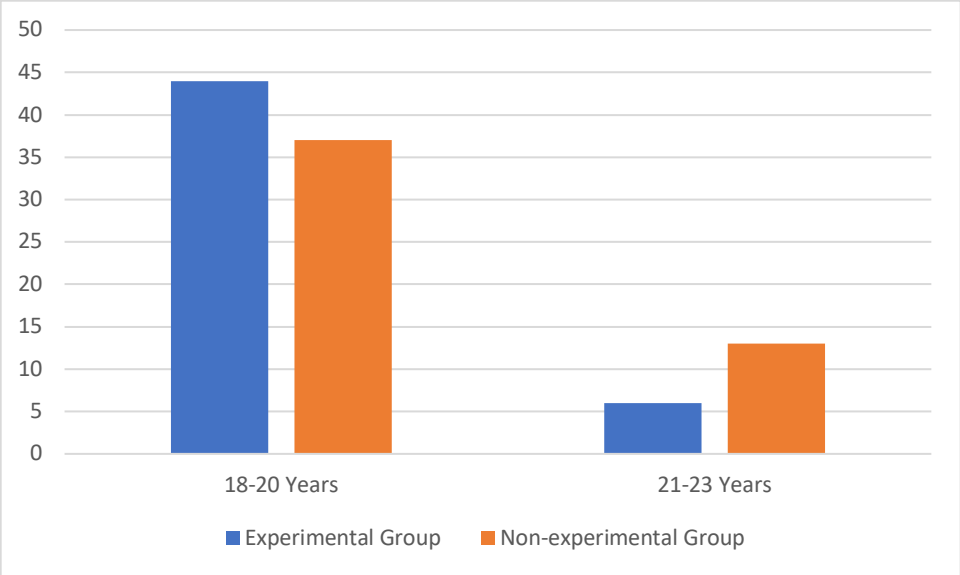

Figure 2 Age Range of Students in Experimental vs. Control Groups

Table 1 Survey Questions Addressing Key Programming Challenges

Challenge	Statement
1. Comprehending Fundamental Concepts	I find it difficult to comprehend the fundamental concepts of programming.
2. Writing Programs	I'm having a hard time writing programs to solve certain problems.
3. Learning Code	Learning and writing code sentences is very Challenging.
4. Error Recognition	Inability to Identify Errors.

4. RESEARCH RESULTS

This section presents the results from both pre-experiment and post-experiment participant surveys, highlighting how the p-values for programming difficulty measures differed between the experimental and control groups. These findings show how the intervention helped shed light on major differences in programming competencies both before and after the experiment.

Table 2: Pre-Experiment on Programming Difficulties Across Experimental and Control Groups

Statement	Experimental Group (Mean)	Control Group (Mean)	p-value
Challenges in Comprehending Fundamental Concepts	2.88	2.98	0.703603
Difficulty writing programs to solve problems	2.88	3.10	0.426286
Learning and writing code sentences is challenging	3.02	2.70	0.245134
Inability to Identify Errors	2.86	2.94	0.770260

This table shows the mean scores and p-values for the various assertions prior to any training intervention.

• **Challenges in Comprehending Fundamental Concepts:**

The experimental group achieved a mean score of 2.88, while the control group attained 2.98 for the same test. The result shows no major difference between groups since the p-value stands at 0.703603. Both groups encounter similar challenges in learning basic programming knowledge.

• **Difficulty Writing Programs to Solve Problems:**

The experimental group again scored lower (2.88) compared to the control group (3.10). The p-value of 0.426286 further supports the conclusion that there is no significant difference in their abilities to write programs to solve problems.

• **Learning and Writing Code Sentences is Challenging:**

The experimental group's mean was 3.02, while the control group scored 2.70. The study results show no significant variation as both groups experienced similar difficulties when writing code, yielding a p-value of 0.245134.

• **Inability to Identify Errors:**

The results indicated that both the experimental and control groups scored nearly the same (2.86 and 2.94, respectively) in detecting errors in code, with no significant statistical difference observed (p-value = 0.770260). The overall results in Table 2 indicated that, before the training, both the experimental and control groups faced similar challenges across different aspects of programming.

Table 3: Post- Experiment on Programming Difficulties Across Experimental and Control Groups

Statement	**Experimental Group (Mean)**	**Control Group (Mean)**	**p-value**
Challenges in comprehending fundamental Concepts	2.68	3.52	0.000853
Difficulty writing programs to solve problems	2.66	3.75	1.67238E-06
Learning and writing code sentences is Challenging	2.70	3.57	0.000457
Inability to Identify Errors	2.50	3.52	1.46576E-07

This table summarizes the results after the training intervention, showing how the mean scores and p-values changed.

• **Difficulty Comprehending Fundamental Concepts:**

The mean of the control group increased to 3.52, while the experimental group mean decreased to 2.68. Experimental group knowledge improved because of the training; this was demonstrated by the p-value of 0.000853 for programming comprehension.

• **Difficulty writing Programs to Solve Problems:**

The experimental group mean was 2.66, whereas the control group was 3.75. The low p-value shows that there is a highly significant difference, 1.67238E-06, meaning that the experimental group has significantly enhanced their capability to write programs after the training.

• Learning and writing code sentences is challenging:

The control group's score increased to 3.57, while the experimental group's score reduced to 2.70. The experimental group's confidence and competence in learning and writing code were positively influenced by the training, as evidenced by the p-value of 0.000457: a significant difference.

• **Inability to Identify Errors:** The experimental group achieved a score of 2.50, while the control group achieved a score of3.52. The experimental group demonstrated a significant improvement in error recognition following the training, as evidenced by the p-value (1.46576E-07). In conclusion, Table 3 demonstrates that the experimental group experienced substantial improvements in all aspects measured following the training, indicating that the training was effective in overcoming the programming challenges they encountered.

4. DISCUSSION

These results of the study provide significant insights into the efficacy of e-learning platforms in improving programming education for diploma-level students at Al-Baha University. The analysis of pre- and post-experiment outcomes reveals a substantial enhancement in the difficulties encountered by the experimental group, which employed e-learning resources, in contrast to the control group, which depended on conventional teaching methods. Initially, the pre-experiment findings indicated that both groups faced comparable challenges in comprehending programming concepts, writing programs, acquiring coding skills, and identifying errors. The lack of significant differences in the mean scores (with p-values ranging from 0.245 to 0.770) indicates that all students struggled with foundational programming skills, a finding consistent with prior studies highlighting the initial challenges faced by diploma-level learners [35]. In contrast, the post-experiment results indicated notable improvements for the experimental group across all assessed challenges, with p-values reflecting a high level of statistical significance (ranging from 0.000457 to 1.67238E-06). For example, the mean score for difficulty in understanding basic concepts decreased from 2.88 to 2.68, while the control group's score increased to 3.52. This substantial shift suggests that e-learning platforms significantly aided the experimental group in mastering foundational programming concepts more effectively than their peers without access to these

resources. The experimental group showed significant enhancements in their ability to write programs and recognize errors, further reinforcing the effectiveness of the e-learning interventions. These results support prior research pointing out that while using e-learning platforms, students apply enhanced problem-solving skills and code more confidently than the students who attended traditional classes [36], [37]. The importance of this study is based on an ability to show that e-learning can solve problems of programming students. As programming is crucial to the modern world education and career, identifying the antecedents of effective programming learning is important and pivotal for the educators and educational institutions of the world. The positive effects found in the present study indicate that incorporating e-learning in curricula can improve the performance of students and increase their self- efficacy on coding skills. The fact that computer-mediated discussions can be used successfully in higher education also has implications for educational practice. integrating e-learning platforms into programming courses to Improve Learning journeys should be considered by Institutions. If students encounter a problem during programming, they can use additional materials and interactive resources made available to them by their instructors to solve a problem. Furthermore, an understanding of how to apply the e-learning tools to the environment will significantly be required by training the faculty. Comparisons with findings from other universities further highlight the effectiveness of e-learning in programming education. For instance, a study at Afyon Kocatepe University in Turkey found that students using blended learning approaches exhibited significant improvements in programming skills and overall course satisfaction [38]. Similarly, research from Zhejiang Normal University in China indicated that students utilizing e-learning platforms outperformed their peers in traditional settings, especially in problem-solving tasks [39]. These comparisons reinforce the notion that e-learning is a valuable pedagogical strategy in programming education, providing a scalable solution to common learning challenges. In summary, this study underscores the significant impact of e-learning platforms on programming education at Al-Baha University. The improvements in student performance observed after utilizing these resources highlight the potential of e-learning to address the challenges faced by programming students. As educational institutions continue to adapt to digital learning environments, the findings from this research can serve as a foundation for future studies and the development of effective teaching strategies in programming education.

6. Conclusion

This study evaluates the effectiveness of e-learning platforms in enhancing the programming skills of diploma-level students at Al-Baha University. The pre- and post-experiment comparison clearly reveals that students using e-learning resources especially W3Schools have shown great improvement in perceived understanding of programming concepts, coding abilities and ability to detect errors. Such research proves that e-learning is a viable supplemental educational tool that resolves multiple difficulties learners encounter in this subject. The improvements noted in the experimental group suggest that e-learning resources, which are easy to interact with, should be part of programming curricula. In the world that is rapidly turning into a digital world, people need programming skills and the educational institutions have to develop the methods of teaching for

this. The results of this study point to the efficacy of e-learning as a means of supporting learning programming, which would enhance students' performance when coupled with enhancement of their self confidence in coding skills. As for the future work, it is proposed to strengthen the research of the effects of e-learning for programming education and look at more e-learning environments that may diversify information helping. Continuing to grow and develop as educators, the incorporation of digital learning becomes enhanced and allows the preparation of students for the solutions and pitfalls presented within the realm of technology.

REFERENCES

[1] T. Korhonen et al., "Finnish teachers as adopters of educational innovation: perceptions of programming as a new part of the curriculum," Computer Science Education, vol. 33, no. 1, pp. 94-116, Jan. 2023, doi: 10.1080/08993408.2022.2095595.
https://doi.org/10.1080/08993408.2022.2095595

[2] X. Du and E. B. Meier, "Innovating Pedagogical Practices through Professional Development in Computer Science Education," Journal of Computer Science Research, vol. 5, no. 3, pp. 46-56, Jul. 2023, doi: 10.30564/jcsr.v5i3.5757.
https://doi.org/10.30564/jcsr.v5i3.5757

[3] K. M. Malik and M. Zhu, "Do project-based learning, hands-on activities, and flipped teaching enhance student's learning of introductory theoretical computing classes?," Educ Inf Technol (Dordr), vol. 28, no. 3, pp. 3581-3604, Mar. 2023, doi: 10.1007/s10639-022-11350-8.
https://doi.org/10.1007/s10639-022-11350-8

[4] C. Mouza, S. Sheridan, N. C. Lavigne, and L. Pollock, "Preparing undergraduate students to support K-12 computer science teaching through school-university partnerships: reflections from the field," Computer Science Education, vol. 33, no. 1, pp. 3-28, Jan. 2023, doi: 10.1080/08993408.2021.1970435.
https://doi.org/10.1080/08993408.2021.1970435

[5] A. Luxton-Reilly, "Learning to program is easy," in Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education, 2016, pp. 284-289.
https://doi.org/10.1145/2899415.2899432

[6] A. Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and discussion," Computer science education, vol. 13, no. 2, pp. 137-172, 2003.
https://doi.org/10.1076/csed.13.2.137.14200

[7] Y. Bosse and M. A. Gerosa, "Why is programming so difficult to learn? Patterns of Difficulties Related to Programming Learning Mid-Stage," ACM SIGSOFT Software Engineering Notes, vol. 41, no. 6, pp. 1-6, 2017. https://doi.org/10.1145/3011286.3011301

[8] A. Gomes and A. J. Mendes, "Learning to program-difficulties and solutions," in International Conference on Engineering Education-ICEE, 2007.

[9] A. V Robins, "12 novice programmers and introductory programming," The Cambridge handbook of computing education research, p. 327, 2019. https://doi.org/10.1017/9781108654555.013

[10] S. Savage and P. Piwek, "Full report on challenges with learning to program and problem solve: an analysis of first year undergraduate Open University distance learning students' online discussions," 2019.

[11] H. C. Jiau, J. C. Chen, and K.-F. Ssu, "Enhancing self-motivation in learning programming using game-based simulation and metrics," IEEE Transactions on Education, vol. 52, no. 4, pp. 555-562, 2009. https://doi.org/10.1109/TE.2008.2010983

[12] J. Bennedsen and M. E. Caspersen, "Revealing the programming process," in Proceedings of the 36th SIGCSE technical symposium on Computer science education, 2005, pp. 186-190. https://doi.org/10.1145/1047344.1047413

[13] T. Jenkins, "On the difficulty of learning to program," in Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, 2002, pp. 53-58.

[14] E. Ng and C. Bereiter, "Three levels of goal orientation in learning," Journal of the Learning Sciences, vol. 1, no. 3-4, pp. 243-271, 1991. https://doi.org/10.1080/10508406.1991.9671972

[15] Y. Qian and J. Lehman, "Students' misconceptions and other difficulties in introductory programming: A literature review," ACM Transactions on Computing Education (TOCE), vol. 18, no. 1, pp. 1-24, 2017. https://doi.org/10.1145/3077618

[16] A. Bough and G. Martinez Sainz, "Digital learning experiences and spaces: Learning from the past to design better pedagogical and curricular futures," Curric J, vol. 34, no. 3, pp. 375-393, Sep. 2023, doi: 10.1002/curj.184. https://doi.org/10.1002/curj.184

[17] J. C. Lapan and K. N. Smith, "'No Girls on the Software Team': Internship Experiences of Women in Computer Science," J Career Dev, vol. 50, no. 1, pp. 119-134, Feb. 2023, doi: 10.1177/08948453211070842. https://doi.org/10.1177/08948453211070842

[18] E. Tereshchenko, A. Happonen, and V. Hasheela-Mufeti, "Barriers for Females to Pursue Stem Careers and Studies at Higher Education Institutions (HEI). A Closer Look at Academic Literature," International Journal of Computer Science & Engineering Survey, vol. 14, no. 1/2/3/4, pp. 01-23, Aug. 2023, doi: 10.5121/ijcses.2023.14401.
https://doi.org/10.5121/ijcses.2023.14401

[19] M. N. Ismail, N. A. Ngah, and I. N. Umar, "The effects of mind mapping with cooperative learning on programming performance, problem solving skill and metacognitive knowledge among computer science students," Journal of Educational Computing Research, vol. 42, no. 1, pp. 35-61, 2010. https://doi.org/10.2190/EC.42.1.b

[20] J. Chetty and D. van der Westhuizen, "Implementing metacognition skills for learners studying computer programming," in EdMedia+ Innovate Learning, 2014, pp. 726-731.

[21] J. Holvikivi, "Conditions for successful learning of programming skills," in IFIP International Conference on Key Competencies in the Knowledge Society, 2010, pp. 155-164. https://doi.org/10.1007/978-3-642-15378-5_15

[22] A. Gomes, L. Carmo, E. Bigotte, and A. Mendes, "Mathematics and programming problem solving," in 3rd e-learning conference-computer science education, 2006, pp. 1-5.

[23] M. N. Ismail, N. A. Ngah, and I. N. Umar, "Instructional strategy in the teaching of computer programming: a need assessment analyses," The Turkish Online Journal of Educational Technology, vol. 9, no. 2, pp. 125-131, 2010.

[24] T. Bhasin, A. Murray, and M.-A. Storey, "Student Experiences with GitHub and Stack Overflow: An Exploratory Study," in 2021 IEEE/ACM 13th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), IEEE, May 2021, pp. 81-90. doi: 10.1109/CHASE52884.2021.00017. https://doi.org/10.1109/CHASE52884.2021.00017

[25] T. T. A. Ngo, T. T. Tran, G. K. An, and P. T. Nguyen, "Students' Perception Towards Learning Massive Open Online Courses on Coursera Platform: Benefits and Barriers," International Journal of Emerging Technologies in Learning (iJET), vol. 18, no. 14, pp. 4-23, Jul. 2023, doi: 10.3991/ijet.v18i14.39903. https://doi.org/10.3991/ijet.v18i14.39903

[26] S. Sharov, S. Tereshchuk, A. Tereshchuk, V. Kolmakova, and N. Yankova, "Using MOOC to Learn the Python Programming Language," International Journal of Emerging Technologies in Learning (iJET), vol. 18, no. 02, pp. 17-32, Jan. 2023, doi: 10.3991/ijet.v18i02.36431. https://doi.org/10.3991/ijet.v18i02.36431

[27] R. Shen and M. J. Lee, "Learners' Perspectives on Learning Programming from Interactive Computer Tutors in a MOOC," in 2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), IEEE, Aug. 2020, pp. 1-5. doi: 10.1109/VL/HCC50065.2020.9127270. https://doi.org/10.1109/VL/HCC50065.2020.9127270

[28] J. Swacha, "DEVELOPMENT AND EVALUATION OF AN INTERACTIVE PYTHON COURSE," Nov. 2018, pp. 456-466. doi: 10.21125/iceri.2018.1094. https://doi.org/10.21125/iceri.2018.1094

[29] V. Karavirta, R. Haavisto, E. Kaila, M.-J. Laakso, T. Rajala, and T. Salakoski, "Interactive Learning Content for Introductory Computer Science Course Using the ViLLE Exercise Framework," in 2015 International Conference on Learning and Teaching in Computing and Engineering, IEEE, Apr. 2015, pp. 9-16. doi: 10.1109/LaTiCE.2015.24.
https://doi.org/10.1109/LaTiCE.2015.24

[30] E. Ponomarenko, A. Oganesyan, and V. Teslenko, "New trends in higher education: massive open online courses as an innovative tool for increasing university performance," International Journal of Economic Policy in Emerging Economies, vol. 12, no. 4, p. 391, 2019, doi: 10.1504/IJEPEE.2019.104635. https://doi.org/10.1504/IJEPEE.2019.104635

[31] J. J. Xu and T. Babaian, "Artificial intelligence in business curriculum: The pedagogy and learning outcomes," The International Journal of Management Education, vol. 19, no. 3, p. 100550, Nov. 2021, doi: 10.1016/j.ijme.2021.100550.
https://doi.org/10.1016/j.ijme.2021.100550

[32] K.-I. Voigt, O. Buliga, and K. Michl, "Democracy in Education: The Case of edX," 2017, pp. 159-170. doi: 10.1007/978-3-319-38845-8_13.
https://doi.org/10.1007/978-3-319-38845-8_13

[33] C. Raibulet and F. Arcelli Fontana, "Collaborative and teamwork software development in an undergraduate software engineering course," Journal of Systems and Software, vol. 144, pp. 409-422, Oct. 2018, doi: 10.1016/j.jss.2018.07.010.
https://doi.org/10.1016/j.jss.2018.07.010

[34]D. I. De Silva, K. A. S. N. Perera, R. A. H. B. Ranasinghe, B. D. Gunawardena, R. R. A. N. N. Jayawardena, and S. Vidhanaarachchi, "CodePedia: Crafting the Ultimate Java Learning Odyssey for Novice Programmers," *Int. Congress on Inf. and Commun. Technol.*, Singapore: Springer Nature Singapore, Feb. 2024, pp. 55–64.

[35] C.-Y. Tsai, "Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy," Comput Human Behav, vol. 95, pp. 224–232, Jun. 2019, doi: 10.1016/j.chb.2018.11.038.

[36] S. I. Malik, R. Mathew, R. Al-Nuaimi, A. Al-Sideiri, and J. Coldwell-Neilson, "Learning problem solving skills: Comparison of E-learning and M-learning in an introductory programming course," Educ Inf Technol (Dordr), vol. 24, no. 5, pp. 2779–2796, Sep. 2019, doi: 10.1007/s10639-019-09896-1.

[37] A. Bernik, D. Radošević, and D. Strmečki, "Research on Efficiency of Applying Gamified Design into University's e-Courses: 3D Modeling and Programming," Journal of Computer Science, vol. 13, no. 12, pp. 718–727, Dec. 2017, doi: 10.3844/jcssp.2017.718.727.

[38] O. Deperlioglu and U. Kose, "The effectiveness and experiences of blended learning approaches to computer programming education," Computer Applications in Engineering Education, vol. 21, no. 2, pp. 328–342, Jun. 2013, doi: 10.1002/cae.20476.

[39] X.-M. Wang and G.-J. Hwang, "A problem posing-based practicing strategy for facilitating students' computer programming skills in the team-based learning mode," Educational Technology Research and Development, vol. 65, no. 6, pp. 1655–1671, Dec. 2017, doi: 10.1007/s11423-017-9551-0.