Afro-Asian Journal of Scientific Research (AAJSR)
المجلة الأفروآسيوية للبحث العلمي
E-ISSN: 2959-6505
Volume 3, Issue 3, 2025
Page No: 241-250
Website: https://aajsr.com/index.php/aajsr/index
SJIFactor 2024: 5.028 ISI 2024: 0.580 معامل التأثير العربي (AIF) 2023: 0.51

Reasons Behind the Failure of Students in The Computer Science Department at The University of Derna

Rabab Algadhy[1*], Huweida Darbi[2], Dalya Qatish[3]
[1,2,3] Computer Science Department, Faculty Science, University of Derna, Derna, Libya

الأسباب الكامنة وراء تعثر الطلاب بقسم الحاسوب بكلية العلوم بجامعة درنة

رباب القاضي[1*]، هويدا دربي[2]، داليا قاطش[3]
[1,2,3] قسم الحاسوب، كلية العلوم، جامعة درنة، درنة، ليبيا

*Corresponding author: r.algadhy@uod.edu.ly

Received: July 04, 2025	Accepted: August 26, 2025	Published: August 30, 2025

Abstract:
Programming skills have become increasingly important in recent years due to the rapid advances in computer science, raising the demand for programmers. However, the complexity of programming poses numerous challenges for inexperienced programmers, contributing to high dropout rates from programming courses. Individual characteristics, including unique neuropsychological traits and the lack of technological infrastructure, particularly in developing countries, are recognized as affecting the accuracy of predicting programming skills outcomes. These are major factors for consideration; hence, we investigate the high student failure rate in the Computer Science Department at the University of Derna, Libya, from the learner perspective concerning the teaching strategies employed by instructors using a qualitative data collection approach. According to the findings, the reasons behind students failure can be divided into three main categories: academic, which is represented by the challenging curriculum's heavy reliance on logic and mathematics as well as its lack of a foundational understanding of programming and analysis; personal and psychological, which is represented by the lack of motivation, work during study sessions, stress from the material's difficulty, and issues with time management; and administrative and environmental, which is represented by the absence of modern labs, technical equipment, and internet networks to support communication and e-learning.

Keywords: Computer Science, Programming, Developing Countries, Failure Rate, Universities.

الملخص
أصبحت مهارات البرمجة ذات أهمية متزايدة في السنوات الأخيرة بسبب التقدم السريع في علوم الكمبيوتر، مما يزيد الطلب على المبرمجين. ومع ذلك، فإن تعقيد البرمجة يفرض العديد من التحديات على المبرمجين عديمي الخبرة، مما يساهم في ارتفاع معدلات الرسوب في تعلم البرمجة. ومن المسلم به أن الخصائص الفردية، بما في ذلك السمات النفسية العصبية الفريدة ونقص البنية التحتية التكنولوجية، وخاصة في البلدان النامية، تؤثر على دقة التنبؤ بنتائج مهارات البرمجة. وهذه عوامل رئيسية يجب مراعاتها؛ ومن ثم، فإننا نحقق في ارتفاع معدل رسوب الطلاب في قسم علوم الكمبيوتر بجامعة درنة، ليبيا، من منظور المتعلم فيما يتعلق باستراتيجيات التدريس التي يستخدمها المعلمون باستخدام نهج جمع البيانات النوعية. ووفقًا للنتائج، يمكن تقسيم أسباب رسوب الطلاب إلى ثلاث فئات رئيسية: الأكاديمية، والتي تتمثل في الاعتماد الكبير للمنهج الدراسي الصعب على المنطق والرياضيات بالإضافة إلى افتقاره إلى الفهم الأساسي للبرمجة والتحليل؛ والشخصية والنفسية، والتي تتمثل في الافتقار إلى الدافع والعمل أثناء جلسات الدراسة والتوتر الناجم عن صعوبة المادة ومشاكل إدارة الوقت؛ والإدارية والبيئية، والتي تتمثل في غياب المختبرات الحديثة، والأجهزة التقنية، وشبكات الإنترنت لدعم التواصل والتعلم الإلكتروني.

الكلمات المفتاحية: علم الحاسب الالي، البرمجة، الدول النامية، معدل الرسوب، جامعات.

Introduction

Over the last few decades, computer science has become one of the most evolving and important disciplines, given its pivotal role in various aspects of life. With the increasing reliance on technology, the need for qualified personnel in this field has become critical. This major requires problem-solving skills, and students are expected to study a wide range of programming modules to develop such skills during four academic years. These modules include problem-solving skills, program design and implementation strategies, and knowledge of programming tools and languages [1].

Computer programming is a fundamental part of computer science and related educational programs. Because of this, anyone interested in studying computer science must become proficient in this essential skill. Students are generally taught how to analyze problems, apply specific methods to develop solutions, and test those solutions after learning about programming and data structures [2]. One of the most important subjects for students studying computer science and information technology is computer programming, which is included in many university curricula [3].

At the University of Derna, students begin learning programming languages in their first year. In the first semester, they are taught programming using C, followed by Python in the second semester. Despite efforts to improve the academic level in the computer science department, there are still worrying indicators regarding student academic progress. A significant number of students are experiencing complications in continuing their studies smoothly, negatively affecting success rates and program retention. The problem lies in the lack of a clear understanding of the reasons behind this difficulty, especially given the interlacing of academic, behavioral, and organizational factors. This requires a scientific analysis based on real data to understand this phenomenon. Hence, this paper investigates the reasons behind student failure who are taking undergraduate courses in the computer science department at the University of Derna.

Our work focused on why students fail in the first two years and do not progress to the next two academic years, leading to poor academic performance, delayed graduation, and probably even dropping out of school. A questionnaire was launched to gather students' opinions and evaluate their educational experiences within the department. The goal is to identify shortcomings and propose paths to improve academic performance and enhance the quality of the educational process. The remainder of this paper is structured as follows. Section 2 gives an overview of related work. Section 3 summaries the study context of the Department of Computer Science at the University of Derna. Section 4 addresses the material and methods used in this study. Section 5 discusses the analysis of the gathered data. Finally, the paper concludes with Section 6.

Related work

Previous studies on information systems courses in developing countries have shown that these nations face a significant shortage of IT skills [4]. This may be due to a limited number of computer science graduates from universities or to a lack of programming education at advanced levels, such as in primary and secondary schools, as is the case in Libya. Because of the skills shortage in developing countries, most information systems projects have been outsourced, which negatively impacts employment opportunities for their citizens. The difficulty of learning programming and the high failure rates in computer science degree programs are two main factors contributing to low interest and high dropout rates, according to research on computer education in developing countries [5].

Few studies about the courses of information systems has addressed the difficulties of studying computer science. For instance, Wall and Knapp [6] investigated how technical information systems courses are perceived as difficult by students and how this perception affects their learning outcomes. They show how students' evaluations of the content's difficulty change as they study it. They either engage in activities that help them manage those challenges or do not, resulting in a detrimental impact on their academic performance with such beliefs.

The natural aptitude of students, before any prior knowledge of programming, has been examined in a study presented by Dehnadi and Bonat [7] which uses a test model to predict students' failure and success. To bolster their argument that teaching programming is futile for those who are likely to fail and useless for those who are guaranteed to succeed. A considerable amount of literature has been published on the teaching strategies and methodologies of programming courses. For example, Sarpong et al. [5] examined the reasons behind students' failure in programming courses from the students' viewpoint concerning the teaching strategies employed by the academic staff. They recommend using a variety of teaching techniques, including laboratory practice sessions, projects, seminars, and tutorials, in addition to lectures, to enable students to have many opportunities and adjust to the best practices. After a decade, another notable study was conducted at Valley View University in Ghana during the 2012-2013 academic year, led by the same researcher, Sarpong et al [8]. The study focused on programming courses in the computer science department, relying on research tools such

as questionnaires and student interviews to identify the best methods and strategies that contribute to improving programming learning and reducing its high failure rates. In this study, laboratory practice was ranked as the most effective teaching method, followed by projects and lectures. Methods such as continuous assessment and exams, e-learning, and field trips ranked least effective. On the other hand, most participants rejected methods that rely on indoctrination, such as instructional instruction and pre-recorded lectures, due to their lack of interaction.

However, there are still limited interdisciplinary studies that have successfully addressed the teaching of technical computing topics. For example, Kelleher & Pausch [9] used storytelling to motivate programming learning. They implemented the Alice environment to create computer graphics movies to stimulate students' interest in learning programming. Similarly, Ibrahim et al.'s [10] study of children's perceptions of using educational games to learn programming found that most students expressed more effective responses and were more likely to advocate for learning through games than traditional methods. Other strategies to motivate programming learning include using faculty evaluation methods and giving feedback on their teaching style [11, 12]. More recently, literature has emerged that integrates different teaching strategies offering promising results for enhancing performance, consolidating concepts, and preventing complexities for learners. For example, an integration of theoretical concepts with immediate practical experience, a combination of pair programming, problem-solving instruction, and e-learning within a virtual environment [13] [14] [15].

The causes of students' learning outcomes, such as passing or failing classes, have been the focus of several studies. Hawi [16] categorized the causal attributions made by undergraduate students in an introductory computer programming course at a university in Lebanon into cultural and domain-specific categories. The ten attributions reported were: "Learning strategy," "lack of study," "lack of practice," "difficulty of the material," "lack of effort," "inappropriate teaching method," "anxiety," "cheating," "time constraints," and "unfair treatment." It has been demonstrated that the relationship between self-efficacy and mental models has a direct impact on students' learning of programming. Wiedenbeck et al. [17] presented a research model demonstrating that students' mental models of programming directly influence their self-efficacy, thereby impacting their success in introductory courses.

However, additional research has concentrated on how various learning styles affect students' performance in programming classes. According to research by Corny et al. [18], students' attitudes toward programming reflect that some learning styles are more suited to learning programming than others. According to research by Tan et al. [19], undergraduate students find that learning programming through lectures alone will not pique their interest as much as learning through examples and a hands-on approach. Personal confidence, perceived motivation, and the utility of programming are additional factors that affect students' success in learning programming. Facey-Shaw and Golding [20] discovered that perceived motivation and utility were not as important as personal confidence in learning programming. Other factors have been identified as significantly influencing first-year students' performance in introductory computer programming modules, such as good academic background particularly in mathematics and science, enthusiasm, prior experience with programming at home or schools, and cognitive skills [21] [22] [23] [24] [25] [26] [27]. However, Dehnadi & Bonat [7] contend that despite the substantial quantity of data on teaching methods and students' reactions to programming language courses, the failure rate has risen over time rather than dropped.

Furthermore, most studies investigating failure rates in programming courses at universities used quantitative approaches and were conducted in industrialized nations. Therefore, there has been a request for further studies to understand the successes and failures of computer programming education in universities in developing countries [8]. The request for additional research underscores the significance of the current study to the field of computer education, particularly in developing nations. In this context, this study employs a qualitative methodology to examine students' failure in programming courses at the University of Derna in Libya, providing a deeper understanding of the underlying causes and explanations that would not be possible with quantitative data. The study's context is discussed in the next section.

Study Context

The Department of Computer Science is one of the key academic departments that keeps pace with rapid technological advancements worldwide. It aims to prepare skilled personnel both scientifically and practically in the fields of computer science and information technology, capable of satisfying labor market needs and contributing to the development of a digital society.

Therefore, the Department of Computer Science was established two years after the founding of the Faculty of Science, during the 1999-2000 academic year, as a division within the Department of Basic Sciences. It became independent in 2004-2005. At that time, the study system was a one-year program, in which students earned a bachelor's degree after completing 102 credit hours across four study phases. In 2022, the program shifted to an open-semester system. The department awards a Bachelor

of Science degree in Computer Science after completing 136 credit hours across eight semesters. The following tables (Table 1 and Table 2) show the programs offered by the Department of Computer Science under both the traditional and open-semester systems. Table 3 shows the optional modules of the open-semester system.

Table 1: Description of the traditional system course for four academic year study in the department of computer science at the University of Derna.

Description of the course			
First year		**Second year**	
Module Name	**Credits**	**Module Name**	**Credits**
Computer Orgnaisation	4	Data Structures	4
Fundamentals of Programming	4	Assembly Language	4
Fundamentals of Mathematics	3	Computer Architecture	3
Discrete Structures	3	Object-Oriented Programming	4
Introduction to Statistics and Probability	3	Systems Analysis and Design	3
General Physics	3	Liner Algebra	3
Arabic Language	2	Number Theory	3
English Language	2	Database Systems	4
Third year		**Fourth year**	
Module Name	**Credits**	**Module Name**	**Credits**
Artificial Intelligence	3	Computer Networks	4
Introduction to Algorithms	3	Intelligent Programming	4
Theory of Automata	3	Software Engineering	4
Computer Security	3	Operating Systems	4
Mobile Applications	4	Data Mining	4
Visual Programming	4	Dissertation	3
Web Programming	4	-	-
Research Methods	3	-	-

Table 2: Description of the open semester course for eight semesters in the department of computer science at the university of Derna.

Description of the course			
First year			
Autumn Semester		**Spring Semester**	
Module Name	**Credits**	**Module Name**	**Credits**
Computer Science	3	Procedural Programming 1	4
General Math 1	4	General Math 2	4
General Statistics	3	Logical Circus	3
General Physics	3	Statistical Methods	3
Arabic Language 1	2	Arabic Language 2	2
English Language 1	2	English Language 2	2
Second year			
Autumn Semester		**Spring Semester**	
Module Name	**Credits**	**Module Name**	**Credits**
Procedural Programming 2	4	Object-Oriented Programming	4
Data Structures 1	3	Data Structures 2	3
Discrete Structures	3	Database 1	4
Computer Architecture	4	Linear Algebra	3
Differential Equations	3	University Skills	3
National Culture	2		
Third year			
Autumn Semester		**Spring Semester**	
Module Name	**Credits**	**Module Name**	**Credits**
Web Programming 1	4	Software Analysis	3
Systems Analysis	3	Operating Systems	3
Database 2	3	Programming Languages	4
Automata Theory	3	Artificial Intelligence	3
Algorithms Analysis and Design	4	Web Programming 2	4

Forth year			
Autumn Semester		**Spring Semester**	
Module Name	**Credits**	**Module Name**	**Credits**
Systems Programming	4	Compilers	3
Research Methods	3	Computer Ethics	3
Computer Security	3	Dissertation	4
Computer Graphics	3	Optional Module 2	-
Computer Networks	4	-	-
Optional Module 1	-	-	-

Table 3: Optional Modules for the open semester system.

Module Name	Credits
Data Mining	3
Image Processing	3
Neural Network	3

Material and methods

This study depends on a grounded theory approach [28] to understand the underlying reasons behind students' failure in programming courses at the computer science department, following the approach presented in [29]. Using the grounded theory approach, researchers were able to find themes and connections between topics by drawing deep insights from the data. Data were collected through a form created on the Gmail platform (Google form) and were analyzed between April and June 2025. To gain a general overview of the primary topics that the participants covered, the responses were carefully read and reread at the start of the procedure. This made it possible to classify related information and insights in the first place. After that, several themes were developed, and emergent ideas were carefully considered. A case study approach was used because it allows for the use of multiple methods to gather data from one or a few entities, such as individuals, groups, or organizations [30]. The study was conducted in the Computer Science Department at the Faculty of Science, University of Derna. It was exploratory in nature, aiming to understand the reasons for students' failure in programming courses so that important lessons can be drawn for decision-makers at both the Computer Science Department and the Faculty of Science.

Subjects

Thirty-five students (ten females and 25 males) from both systems participated, representing the first, second, third, and fourth year at the Department of Computer Science, Faculty of Science, University of Derna. According to Table 4, this represents approximately 47% of the total targeted students. Their ages ranged from eighteen to thirty years and up. Thirty-one pupils have a general secondary school certificate (science stream), two have a private secondary school certificate, and the other two have graduated from college.

Table 4: Demographics of subjects

Description		Total
Age	18-20	16
	20-25	15
	25-30	2
	30 and above	2
Gender	Female	10
	Male	25
Academic qualification	General Secondary School	31
	Private Secondary School	2
	High college	2
	Secondary college	0
Academic stage	First year	10
	Second year	9
	Third year	9
	Fourth year	7

Research tool

In this study, the questions were crucial, as they enabled us to observe the difficulties students face while learning programming at different stages of their education. The questionnaire consists of thirty questions, as shown in Appendix 1. Unlike the work presented by Dasuki and his team [29], some questions (Questions 5 and 6) were added to examine the extent to which work during study and reliance on parents for instruction impacted students' academic performance. At the end of the questionnaire, participants were asked to raise any other issues they considered relevant. We did not focus on the opinions of academic members, as we did not target them with the questionnaire.

Results and discussion

The process started with a careful reading and rereading of the participants' responses to get an overview of the main themes discussed. A qualitative method was used, which depends on texts and interpretation, and involves interviews or open-ended questions. Only 46% of the department's students were included, which we deemed a good percentage and covered more than one aspect of the students' perspectives. Most of the students were male, reflecting the majority gender in the department. This primarily allowed for the categorization of similar material and perspectives. Next, a set of themes was developed, paying close attention to emerging themes. The following paragraphs highlight significant quotes and relevant themes from the responses to illustrate the coding process. This section presents the results of the case study analysis, along with a discussion of the findings, emphasizing the challenges faced by students in the Computer Science Department at the University of Derna in understanding programming courses. The following points address the main themes that emerge from the data related to the reasons for students' failure or stumbling. These themes include:

Working While Studying

Students' responses to the fifth question of the survey revealed that 28 out of 35 students, or 80%, work while studying, as shown in Figure 1. This may affect their academic performance, as they lack sufficient time to study and focus on solving programming problems.

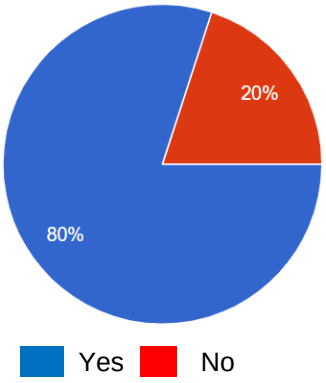

Figure 1: Percentage of students who work while studying.

Lack of motivation

Motivation can be defined as the inspiration that comes from within a person. This motivation typically stems from the enjoyment one feels when performing a task. In our study, students' intrinsic motivation to solve and evaluate programming issues was adversely affected by several contextual factors. Many students agreed that the reasons for their failure in programming were due to their lack of enjoyment due to a lack of interest, and concentration, confirming that some of them were working while studying, which affected their academic performance, as illustrated by some of the quotes below:

"Anyway, I will not focus"

"There is no difficulty, but there is a lack of interest from the students "

In contrast, students attributed their lack of motivation to the classroom atmosphere between the lecturer and the student. Many felt the classroom atmosphere was boring and quiet, making them feel sleepy and unengaged with the lecturer due to traditional teaching methods. As a result, students attended just to avoid losing grades. This led to a weak relationship between lecturers and students, and consequently, a lack of interest and ambition to learn programming. The following quote illustrates the impact of a weak lecturer-student relationship on students' motivation to learn programming:

“Honestly, it is boring now. I attend only so that I do not lose grades, and I know very well that if I am absent, no one will send me what I missed, so I attend the lecture even if I do not want to, and I feel bored, tired, and sleepy so that I do not fail.”

Expectations of the future

Future expectations reflect the predicted worth of studying programming. Few computer science students find programming unhelpful in their degree program. While many students believe it is useful for future career, others are distracted by the lack of emphasis on learning a specific programming language. This demonstrates a lack of experience with learning programming concepts and developing problem-solving abilities. One student noted:

"Unfortunately, I am not good at any programming language due to the dispersion of programming languages and the work environment that changes in each semester. Yes, most languages are related, but the only problem is that whenever I try to master a certain language, I find that in this semester I will not use this language during my studies, which makes me forget to work with it."

The above quote demonstrates that many students believe that to master programming, they must focus on a single language and learn it for several years. From this perspective, they emphasize the need to focus instruction on a single programming language, which underscores their lack of understanding of the concept of programming problem-solving, as illustrated in the quote below:

"Honestly, it's related, but the difference is only in terms of distraction and not being able to master a certain language."

In conclusion, students' misconception that learning programming is about focusing on and mastering one programming language has led to a lack of effort and a lack of passion for learning programming.

Anxiety

Anxiety is defined as a state of disquiet, tension, or concern over something that has an unclear consequence. The majority of computer science students in this study appeared to have programming anxiety, which harmed their academic performance. Student responses indicated that the department's lack of prior knowledge of problem-solving contributed significantly to their anxiety. Some students spoke of being distracted by the lack of focus on learning a single programming language, as illustrated in the quotes below:

"The difficulty lies in the project ideas for those who don't have a good foundation, like me. It also lies in the practical exams, which can be fraught with fear and confusion. Perhaps the most common difficulty is the lack of focus on learning a particular language throughout the study period, which leads to distraction and a lack of concentration."

"The computer science courses in the department require a high level of analytical and logical thinking, but they are designed to gradually develop these skills in students."

Furthermore, gathered responses revealed that students' lack of prior programming knowledge and poor mathematics background significantly contributed to their anxiety, as some students emphasized in the quotes below:

"We don't understand the basics from the very beginning of teaching in schools, and I don't expect it that way."

"I have difficulty with math subjects."

"I find the most difficult subject to be general math at all levels, but the rest is easy for me to study."

Peer influence

Most students confirmed that when they asked their peers about computer science, they were told it is a difficult subject, which dampened their enthusiasm for the program and created an involuntary sense of difficulty before even starting the course. One student expressed regret for choosing computer science due to the irritation of individuals who inquire about the major. Another reported that he focuses on studying programming while everyone around doing so, it is believed that it was difficult to study individually as illustrated in the following responses:

"It's hard and you need to be good in English and the subject is not easy so I feel it's hard before I even start studying."

"Honestly, every time I talk to someone, I regret entering this major."

"If they're focused on studying, you'll automatically start focusing with them. If they're not focused, studying individually becomes difficult."

Most students emphasized the lack of support from their friends due to the difficulty of the major, as they described it. Others expressed frustration at the lack of communication among them, such as social media groups for exchanging opinions and discussing specific issues., as in the quote below:

"*They don't help, neither calls, nor messages, nor even groups. "*

"When I find someone who is excellent in the field, but he is an introvert who does not have any social relationships"

Lecturer skills and behavior

Students reported that faculty members possess the required skills to teach them programming properly. They also noted the lecturers' success in making programming enjoyable and motivating students to learn it, as illustrated in some of the quotes below:

"Yes, the lecturer has an engaging style of presenting the scientific content, which makes the lecture enjoyable and useful."

"Yes, when there is interaction, discussion, and exchange, and when the lecturer provides us with information outside the curriculum related to practical life or gives us advice or anything else we can benefit from, it makes the lecture enjoyable and makes you want to come to the university for the lecturer."

"The lecturer makes a great effort to convey the information."

Weak practical capabilities

Computer science students rely heavily on the practical side to acquire programming and technical skills. However, in the absence of equipped computer labs or modern devices capable of running the required software, or original and updated software, the student is forced to rely on ineffective alternative methods, such as theoretical learning only, which greatly weakens their practical skills, as mentioned in the quote below:

"I propose a specialized workshop presented by industry experts, which will help network with companies for collaboration opportunities and greater volunteer participation. In addition, a wonderful organisation for a new computer lab assistant with two high-performance computers and good lighting conditions, to work effectively and achieve further progress."

Another problem is internet outages or slow network connections. Most computer technologies rely on the internet, whether for research, practical training on cloud tools, or access to repositories. Poor connectivity deprives students of modern practical training and participation in competitions or global communities. Lack of resources has an impact not only on academic performance, but also on students' professional identity, innovation, and self-confidence. The approach is to increase funding while simultaneously making the most use of available resources, giving digital support and open-source alternatives, and fostering interactive self-learning.

In conclusion, this study examined the reasons for student failure in the Computer Science Department at the Faculty of Science, University of Derna. Using a case study approach, the study identified low self-motivation, poor future expectations, anxiety, peer influence, and limited practical work environment as challenges leading to a high failure rate in the Computer Science Department. Students lack focus on their studies due to working while studying, which negatively impacts their classroom performance. They also lack a clear understanding of the importance of learning programming in their studies, as well as poor problem-solving, logical, and analytical thinking skills. The study results also showed that many students find programming boring and difficult to understand. The study also showed that some lecturers typically assume students have programming experience. However, many students indicated that they had no prior programming experience before enrolling at the university and therefore expected programming to be taught from the basics. Students strongly criticized their lack of fundamental programming experience, consistently citing it as an explanation for their difficulty understanding the programming concepts taught by lecturers. Their poor background in mathematics also contributed to the high failure rate, as they believed that mathematics was not related to computer science. This stemmed from insufficient information about computer science.

Conclusion

This study aims to investigate the most prominent factors that lead to the academic failure of students in the Computer Science Department at the Faculty of Science, University of Derna, whether in terms of low grades, delayed graduation, or poor practical and theoretical performance. The results indicate that the reasons fall into three main categories: academic represented in difficult curriculum and its heavy reliance on mathematics and logic and weak basic background in programming and analysis; personal and psychological represented in lack of motivation, work while studying, stress resulting from the difficulty of the material and problems with time management; administrative and environmental represented in lack of up-to-date laboratories and technical equipment and internet networks to facilitate communication and e-learning.

It is recommended that developing the department's curricula to keep pace with modern developments and providing a stimulating learning environment based on practical application and interactive learning. The department's infrastructure should also be developed to include modern laboratories, internet connectivity, and cloud computing to facilitate communication between students and faculty members. Students who experience programming anxiety often feel inadequate. To improve their self-efficacy, lecturers should ensure an engaging, interactive, calm and non-competitive

environment. They should also provide adequate support and guidance to students in correcting programming errors, as well as help them feel comfortable.

The above recommendations are a first step toward improving the graduation success of students in the Computer Science Department at the University of Derna, contributing to Libya's economic growth. In proposing potential for future work, the limitations of this study are acknowledged. The data collected in this research were from the student perspective, and future research in this area could combine the perspectives of both students and academic members in the department. However, there is scope for a more in-depth study based on the current findings to provide a deeper understanding of other challenges by expanding the study to include all students enrolled in computer science programs at universities across the country. The study's findings cannot be generalized because students at the University of Derna may have different academic levels than students at other universities.

References

[1] K. Ala-Mutka, “Problems in learning and teaching programming-a literature study for developing visualizations in the Codewitz-Minerva project,” *Codewitz needs analysis,* vol. 20, 2004.

[2] Khaleel, M., Jebrel, A. and Dunia M. Shwehdy 2024. Artificial Intelligence in Computer Science. *Int. J. Electr. Eng. and Sustain*. 2, 2, 01–21, 2024.

[3] S. a. S. V. Mohorovicic, “An overview of computer programming teaching methods,” *Central European Conference on Information and Intelligent Systems,* p. 47, 2011.

[4] D. Dada, “The Failure of E-Government in Developing Countries: A Literature,” *The Electronic Journal of Information Systems Developing Countries,* pp. 1-10, 2006.

[5] K. Sarpong, J. Authur and . P. Owusu, “Causes of Failure of Students in Computer Programming Courses: The Teacher Learner Perspectve,” *International Journal of Computer Applications,* pp. 77, 12, 2003.

[6] J. Wall and K. Knapp, “Learning Computing Topics in Undergraduate Information Systems Courses: Managing Perceived Difficulty,” *Journal of Information Systems Education,* pp. 245-259, 2014.

[7] S. Dehnadi and R. Bonat, “The Camel Has Two Humps,” 2006. [Online]. Available: https://www.eis.mdx.ac.uk/research/PhDArea/saeed/paper1.pdf.

[8] K. A.-M. a. A. J. K. a. A. P. Y. O. Sarpong, “Causes of failure of students in computer programming courses: The teacher-learner Perspective,” *International Journal of Computer Applications,* pp. 27-32, 2013.

[9] C. Kelleher and R. Pausch, “Using Story Telling to Motivate Programming,” pp. 59-64, 2007.

[10] R. Ibrahim, R. Yusoff, H. Omar and A. Jafar, “Students’ Perceptions of Using Educational Games to Learn Introductory Programming,” in *Computer and Information Science*, 2011.

[11] W. Ng, “The Impact of Peer Assessment and Feedback Strategy in Learning Computer Programming in Higher Education,” 9 2012. [Online]. Available: http://iisit.org/Vol9/IISITv9p017-027NG039.pdf.

[12] J. Sitthiworachart and M. Joy, “Web-based Peer Assessment in Learning Computer Programming,” in *In Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies*, 2003.

[13] W. M. a. A. R. B. Kunkle, “The impact of different teaching approaches and languages on student learning of introductory programming concepts,” *ACM Transactions on Computing Education (TOCE),* pp. 1-26, 2016.

[14] S. a. T. S. K. a. A. M. a. H. M. Munawar, “Move to smart learning environment: Exploratory research of challenges in computer laboratory and design intelligent virtual laboratory for eLearning technolog,” *EURASIA Journal of Mathematics, Science and Technology Education,* pp. 1645-1662, 2018.

[15] D. Pllana, “Combining teaching strategies, learning strategies, and elements of super learning principles,” *Advances in Social Sciences Research Journal,* pp. 288-301, 2021.

[16] N. Hawi, “Causal Attributions of Success and Failure Made by Undergraduate Students in an Introductory-Level Computer Programming Course,” in *Computers & Education*, 2010.

[17] S. Wiedenbeck, D. LaBelle and V. Kain, “Factors Affecting Course Outcomes in Introductory Programming,” in *In Proceedings of the 16th Annual Workshop of the Psychology of Programming Interest Group*, 2004.

[18] M. Corney, D. Teague, and R. Thomas, “Engaging Students in Programming,” in *Proceedings of the Twelfth Australasian Conference on Computing Education*, 2010.

[19] P. Tan, C. Ting, Y. Hoo and S. Ling, “Learning difficulties in programming courses: undergraduates' perspective and perception,” in *2009 International Conference on Computer Technology and Development*.

[20] P. Facey-Shaw and P. Golding, “Effects of Peer Tutoring and Attitude on Academic Performance of First Year Introductory Programming Students,” in *n Proceedings of 35th ASEE/IEEE Frontiers in Education Conference, Indianapolis*, 2005.

[21] J. Bennedsen and M. Casperson, “Failure Rates in Introductory Programming,” in *ACM SIGCSE Bulletin*, 2007.

[22] P. Byrne and G. Lyons, “The effect of student attributes on success in programming,” in *Proceedings of the 6th annual conference on Innovation and technology in computer science education*, 2001.

[23] P. F. Campbell and G. P. McCabe, “Predicting the success of freshmen in a computer science major,” *Communications of the ACM,* pp. 1108-1113, 1984.

[24] H. a. A. L. a. G. S. a. W. E. a. M. E. a. D. J. a. M. J. Holmegaard, “Feeling the weight of the water: A longitudinal study of how capital and identity shape young people’s computer science trajectories over time, age 10--21,” *Computer Science Education,* pp. 238-266, 2025.

[25] S. McGee, R. I. Greenberg, L. Dettori, A. M. Rasmussen, R. McGee-Tekula, J. Duck and E. and Wheeler, “An Examination of Factors Correlating with Course Failure in a High School Computer Science Course.,” 2018. [Online].

[26] C. a. M. A. I. a. R. M. A. a. L. C. Navarro-Cota, “Individual differences in computer programming: a systematic review,” *Behaviour and Information Technology,* pp. 357-375, 2025.

[27] N. Rountree, J. Rountree and A. Robins, “Predictors of success and failure in a CS1 course,” in *ACM SIGCSE Bulletin*, 2002.

[28] J. M. a. S. A. Corbin, “Grounded theory research: Procedures, canons, and evaluative criteria,” *Qualitative sociology,* pp. 3-21, 1990.

[29] S. Dasuki and A. Quaye, “Undergraduate students’ failure in programming courses in institutions of higher education in developing countries: A Nigerian perspective,” *The Electronic Journal of Information Systems in Developing Countries,* pp. 1-18, 2016.

[30] I. Benbasat, D. K. Goldstein and M. Mead, “The case research strategy in studies of information systems,” *MIS quarterly,* pp. 369-386, 1987.