LJS Vol. 29, No. 1 (2026) 1 - 9

The Libyan Journal of Science
University of Tripoli Vol. 29, No. 1
https://uot.edu.ly/journals/index.php/ljs

Introductory Computer Programming Course at the University of Tripoli: Learning Difficulties and Impact of Preferred Study Materials on Students

Adnan Elsherif [1*], Ali Aburas [1], Zainab Ahmed [1]

1. Computer Science Department

Faculty of Science, University of Tripoli

Tripoli - Libya

Corresponding author: **Adnan Elsherif* adnan.sherif@uot.edu.ly

ARTICLE INFO

Article history:

Received *25/1/2025*

Received in revised form *9/12/2025*

Accepted *14/02/2026*

ABSTRACT

Notably, many students entering the University of Tripoli fail or drop out of the introductory programming course. Freshmen face various difficulties in introductory computer programming courses. Some common challenges they face include problem-solving, learning syntax, and editing and writing the code. In this study, 224 students in the computer science department at the Faculty of Science were enrolled in an introductory Python programming course. The study aims to identify and understand the challenges freshmen face in an introductory computer programming course. Furthermore, we explored the types of materials students prefer in an introductory computer programming course. We also explored the impact of the preferred material on understanding the programming concepts. The statistical analysis of the data gathered from students identifies and ranks the main difficulties they face. We found that understanding the problem in the programming assignment was the major difficulty students faced. We further explored the relationship between the difficulties the students face, and the materials used to study the programming course. The results also support our hypotheses of correlations between student difficulty and private courses provided outside the university. The results show no relationship between student difficulty learning and private courses provided outside the university. Finally, we concluded with recommendations on improving the material contents and adapting the study material that students prefer in teaching introductory computer programming courses.

Keywords: Introductory Programming; Learning Difficulties; Freshmen Students; Learning Materials.

1. Introduction

Clearly, many students have failed or dropped out of the Introduction to Programming course in recent years. This could be a result of many factors, such as the difficulties with the topics of programming languages, e.g., control statements and variable types [1], [2], and the lack of skills in developing the materials, e.g., lecture notes, and assignments, that the students need to study the course. As a result, students face different challenges when learning programming. The course materials and content must be adjusted to address these challenges that impact the student's success. To the best of our knowledge, these problems have not been addressed at the university.

This study identifies factors that affect students' success in the Introduction to Programming course. The aim of the study is to answer the following research questions:

Q1: What are the main difficulties faced by the students when learning an Introduction to Programming Course?

Q2: What are the topics the students have the most difficulty understanding?

Q3: What type of material helped you understand the issues in programming?

To answer the research questions, a survey is prepared. The survey focuses on identifying the difficulty level of each activity in the programming learning process, the topics and their difficulty levels, the students' preferred learning methods, and the materials used in an introductory programming course.

The survey data are analysed to answer the research question. The study's results make recommendations for teaching Introduction to Computer Programming courses. These recommendations can improve the course material for introductory programming courses.

In the next section, we provide a brief overview of related research trends in the field. The methodology used and the survey are explored in detail in the following section. Next, a detailed analysis of the data collected is presented. We conclude with the results and recommend future work in the final section.

2. Similar works

During the first semester at the university, every computer science freshman takes a foundational programming course [3], [4]. Several studies [5], [1], [6] found that learning and teaching programming are difficult tasks that require a lot of enthusiasm, focus, and dedication from everyone involved. Due to the high failure and dropout rates in introductory programming courses [7], [8], studies focus on investigating, identifying, and analysing the different challenges that students have when attempting to study introductory programming courses [1], [3], [5] [6], [9], [10].

Several Studies have identified common causes of difficulties, such as a lack of students' skills, students' poor ability to solve problems, a failure to comprehend the ideas, a lack of understanding of inappropriate learning environments, misinterpretation of complex elements, difficulty writing algorithms and designing solutions and difficulty debugging programs, etc. [1], [6], [7], [10], [11] and [12].

According to the findings of Majalin et al. [9] on Difficulties in Learning Java Programming for Secondary School, students had trouble with a range of programming subtopics, from simple to complicated concepts. A few difficulty patterns have already been identified in [2]. Some of these are the challenge of working with functions, the scope of variables and the logic behind passing and returning parameters.

Qian & Lehman [13], conducted a survey to investigate misconceptions and students' challenges

when learning programming, and they found students have difficulties in syntactic, conceptual, and strategic knowledge. Program control concepts such as selection and iteration structures have been shown to be among the hardest for beginners to understand, according to [12].

Regarding the materials students use in introductory programming courses, choosing the right materials can significantly impact learning to program [14]. However, another study shows that students prefer learning in a classroom setting with lectures [9]. Most undergraduates agreed that they can learn programming more effectively in practical or laboratory sessions [1], [11] and [14].

Research on the challenges students face when learning programming is ongoing, with the goal of assessing the environment, materials, and available resources to develop a suitable learning paradigm. Several studies have suggested assessment tools for use in the teaching and learning process to alleviate the challenges students have when learning the basics of programming [1], [5], [11], [14] and [15].

3. Methodology and Survey

All freshmen students at the Faculty of Science must study the Introduction to Computer Programming course (CS100). The course objective is to teach students how to understand and solve problems using a programming language [6]. The programming language used in the course is Python, chosen for its simplicity, which makes it ideal not only for beginner programming in the computer science major but also for other majors [1], [2].

To collect information on students' difficulties learning their first programming language and the factors that affect these difficulties, a survey was distributed to freshmen at the Faculty of Science. The survey was distributed on paper to students during the course final exam, and they answered the questions anonymously. Then, they left the survey answers on a table at the exit of the exam room.

3.1 Survey Content

The survey is structured in four parts. The first step is to identify the participants' gender and age. The three consecutive parts, each corresponding to a different research question, are as follows:

Q1- What are the main difficulties faced by the students when learning an Introduction to Programming Course?

The research question aims to understand the difficulty level of each programming learning activity. The activities follow the course structure and represent the steps to learning any introductory programming course [16], the activities include: (a) Understand the problem of the programming assignment; (b) Conceiving the solution idea; (c) Designing a flowchart or writing pseudocode for the solution; (d) Use the syntax of the programming language; (e) Writing and running the program; (f) Tracing the code and finding the values of the variables and the outputs of the program; (g) Use of the Integrated Development Environment (IDE).

Each activity is rated by the student on a scale from 1 to 5, with 5 Extremely difficult and 1 Extremely easy. The students are instructed to answer the survey using the scale as follows: Scale 1-Extremely Easy: The student did not face any problem in understanding and using this activity or topic; Scale 2-Easy: Some problems have been faced by the student in understanding and using the topic, however, little effort was needed to overcome these difficulties; Scale 3-Moderate: The difficulty faced by the student was solved by the student, using the available material and other means, however, without seeking help from

others; Scale 4- Difficult: The student could not understand and use the topic by themselves and needed assistance from the teacher or other students; Scale 5-Extremely Difficult: Despite effort by the student to seek help to understand the topic the student continues to not understand the topic and how to use it.

Q2- What are the topics the students have the most difficulty understanding?

Here, the question addresses topics related to a programming language. The objective is to identify which programming language constructs students have more difficulty understanding. The student needs to rate, using the same scale in the previous question, the following topics: (a) flowcharts; (b) pseudocode; (c) Variables, types, and scope; (d) Assignment statements, expressions, and order of operands; (e) Conditional Statements; (f) Loops; (g) Lists or Arrays; (h) Function definition and function call.

Q3- What type of material helped you understand the issues in programming?

We explored 8 types of material used by students in the course. The selection of these materials was based on instructors' experience teaching the course, discussions with students, and the materials prepared and made available to students by the university. For each of these types of materials, Students are given the option to select either: Not used, if they did not use the material while learning the course; Not Helpful, if they did use the material but found it not to be useful in understanding the course; Helpful, if the students used the material and found it to be useful in understanding the course; Extremely Helpful, case the material was the most effective in understanding the course. The material rated by students are: (a) Lecture Notes; (b) Books and other printed material; (c) Examples and assignment solutions; (d) Audio and Video provided by the university; (e) YouTube and third-party video material; (f) Attending the lectures in class and asking questions; (g) Private programming courses outside the university; (h) Programming courses online.

3.2 Dataset of Students

A total of 224 students answered the survey correctly. The gender distribution is shown in Figure 1, 105 Male students have taken the survey against 119 Female students. This shows almost even gender distribution of the participants.

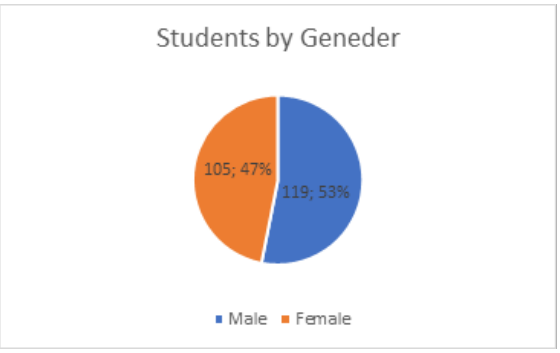

Figure 1 Gender distribution

Furthermore, the age of first-year students is expected to range from 18 to 20 years. We observe from **Figure 2**, that 62% (138 Students) fall within this range among freshmen. Whereas 35% (78 Students) are in the age range of 21 to 23.

In the next section, the data is analysed to answer the survey questions stated in the previous section. We conduct several analyses and further explore these questions to better understand the difficulties students face.

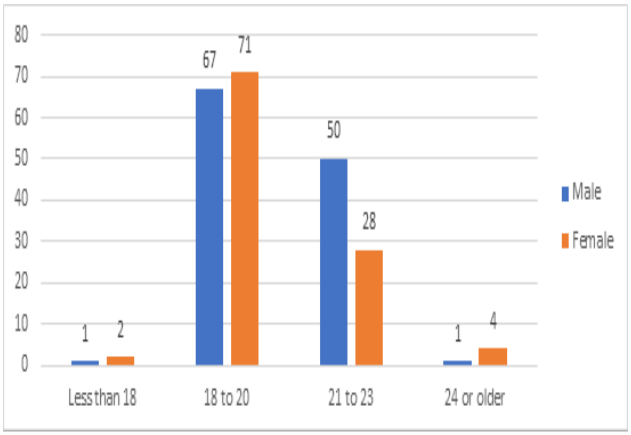

Figure 2 Student distribution by age and gender

4. Results

The survey data are analysed quantitatively to identify the difficulties students face. We further cross-reference these difficulty levels with students' preferred study materials to assess how the chosen material affects learning a new programming language.

solution idea (D2), designing a flowchart or pseudocode (D3), using the syntax of Python (D4), writing and running the program(D5), tracing the program and finding the values of variables (D6) and using the integrated development environment (IDE)(D7).

Table 1 shows the median, mean, and standard deviation of the students' main difficulty in learning programming. From the data shown in Table 1, we observe that understanding the problem of the programming assignment (D1), conceiving the solution idea (D2), and using the integrated development environment (IDE)(D7) are the main difficulties that the student faced during learning programming, with mean values above 3.0. We also observe that the D2 has a standard deviation of 0.79, indicating that students agreed this topic is the most difficult. However, designing a flowchart or pseudocode (D3) is the least difficult activity students face (mean = 2.55, standard deviation = 0.98).

Table 1 Analysis of the difficulty in understanding programming language course contents

Statistical Results	D1	D2	D3	D4	D5	D6	D7
Median	3	3	3	3	3	3	3
Mean	3.05	**3.09**	**2.55**	2.88	2.86	2.87	3.05
Standard Deviation	0.82	0.79	0.98	1.00	0.91	0.90	1.04

4.1 Analysis of the main difficulties faced by the students when learning an Introduction to Programming Course.

In this section of the survey, the students answered the difficulties they faced in understanding the problem of the programming assignment (D1), conceiving the

4.2 Analysis of difficulty in understanding programming language course contents

In the previous section, we demonstrated the students' difficulties in understanding, learning, and applying

Table 2 Analysis of difficulty in understanding programming language course contents

Statistical Results	T1	T2	T3	T4	T5	T6	T7	T8
Median	2	2	3	3	3	3	3	4
Mean	2.30	2.31	2.59	2.84	2.74	2.94	3.18	3.53
Standard Deviation	0.94	0.98	0.88	0.92	0.90	0.95	0.92	1.01

the skills and activities needed to learn a programming language. In this section, we explore various programming language topics and examine students' difficulties with each.

As we mentioned earlier, the first-year students learn in the introductory programming course the following topics: how to design Flowcharts (T1), Pseudocode (T2), understand variables, variable types, and variable scope (T3), assignment statements, mathematical and logical expressions, and order of operands in an expression (T4), conditional statements (T5), loops (T6), lists (T7), function definition and function call (T8).

Table 2 shows the results of the statistical analysis of data on students' difficulties with programming content. The function definition and function call topic (T8) is the most difficult among the topics, with a mean of 3.53 and a standard deviation of 1.01. Most students find the remaining contents (T1 to T7) moderately difficult. We can also observe that the standard deviation of all topics is approximately the same. This indicates that students' perceptions of the difficulty of programming topics are similar, with a slight increase for the most difficult topic, Function definition and function calling (T8).

4.3 What type of material helped the students understand the topics in programming?

In this section, we explore the material students use to study for the programming course and its effectiveness from the students' perspective. We explored 8 types of material used by students in the course. Lecture Notes (M1), Books and other printed material (M2), Examples and assignment solutions (M3), Audio and Video provided by the university (M4), YouTube and third-party video material (M5), Attending the lectures in class and asking questions (M6), Private programming courses outside the university (M7), and Programming courses online (M8).

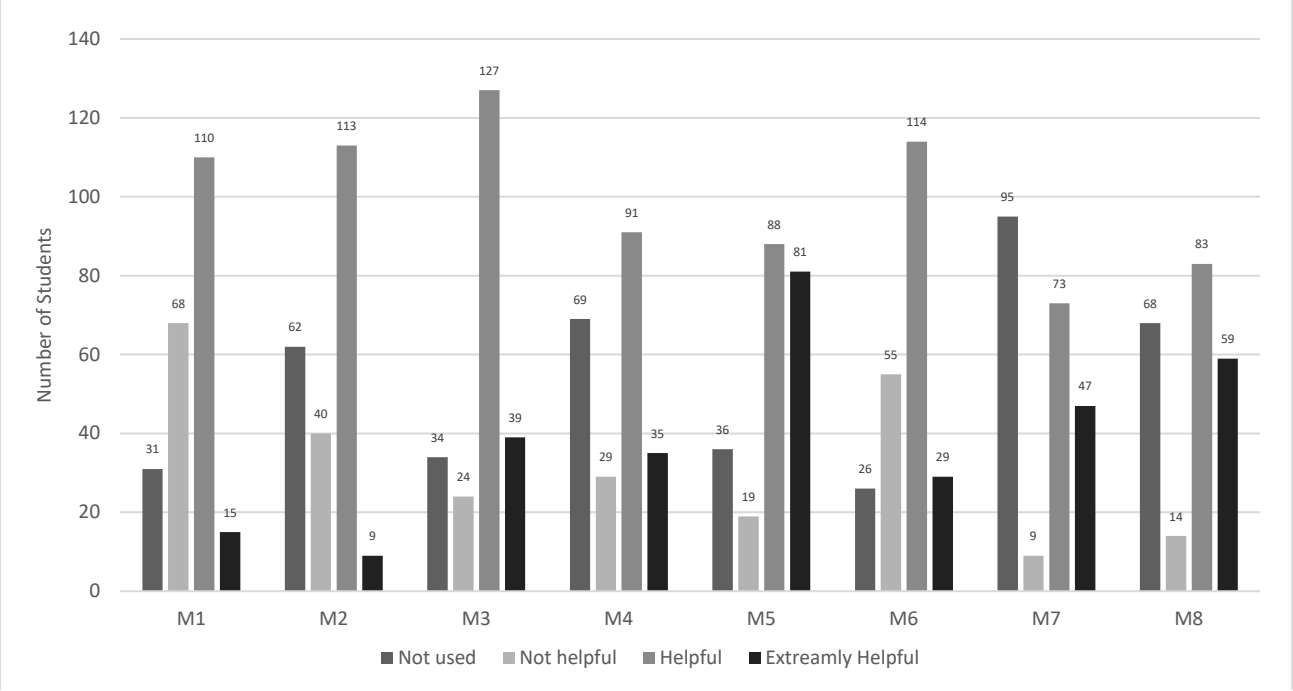

Figure 3 Analysis of the material used by students.

Figure 3 presents the statistical results on the types of materials students used. From Figure 3, 166 students found examples and assignment solutions (M3) to be either helpful (127) or extremely helpful (39), making this method the most used and most

helpful. However, 34 students did not use the examples and assignment solutions. An alarming result is that, for Lecture Notes (M1), 68 students rated it as used and found it not helpful, making this method the least helpful among the studied methods. This result is a call to revise the teachers' lecture notes.

4.4 Correlations

This section explains how correlations can help us better understand survey results. Correlation compares and shows the strength of the relationship between two or more survey questions.

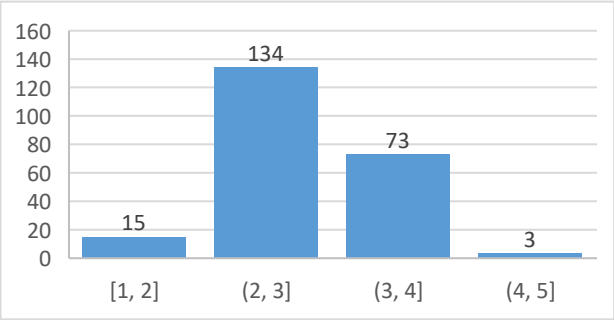

Figure 4 Student Difficulty Level distribution

Moreover, we calculate the mean for each student's answers to research question Q1. The mean value indicates the student's difficulty level with the topics addressed in Q1. As illustrated in Figure 4, 15 students have a difficulty level ranging from 1 to 2 inclusive; these students find most of the topics easy or extremely easy, and they represent 6.6% of the total students. Most students (134 Students) have difficulty level 2 or higher; these students find most topics moderate to easy, representing 59.8% of students. Meanwhile, 73 Students have a difficulty level higher than 3 but lower than or equal to 4; they represent 32.5% of the total students. This group finds the difficulty of learning programming to be moderate to difficult. The last group, representing 1.1% of total students, finds that learning a programming language is extremely difficult, with their difficulty level ranging from 4 to 5.

Table 3 shows the correlations between students' difficulty levels and the types of materials that helped them understand the programming topics during the course. The students who used examples and assignments (M3) had less difficulty learning programming. This is reflected in the negative correlation between (M3) using examples and assignment solutions and the student difficulty level (r = -0.2).

In addition, there is no correlation between the student's difficulty level and attending private courses outside the university (M7) (r=0.01), which comes as

Table 3 Correlation between different study materials and student difficulty

	M1	M2	M3	M4	M5	M6	M7	M8	Student Difficulty
M1	1.00								
M2	0.23	1.00							
M3	0.27	0.33	1.00						
M4	0.30	0.16	0.37	1.00					
M5	0.10	0.15	0.27	0.26	1.00				
M6	**0.45**	0.23	0.31	0.37	0.27	1.00			
M7	0.04	0.13	0.16	0.24	0.25	0.14	1.00		
M8	0.07	0.10	0.10	0.15	**0.41**	0.05	0.39	1.00	
Student Difficulty	-0.17	-0.05	**-0.20**	-0.12	-0.18	-0.16	**0.01**	-0.17	1.00

a surprise because there is a belief that students prefer to attend private courses outside the university.

An interesting correlation arises from the **Table 3** We notice that students who prefer lecture notes (M1) also prefer attending lectures and asking questions (M6) (r=0.45). On the other hand, students who find YouTube and third-party videos (M5) helpful also find online courses (M8) helpful (r = 0.41). This leads us to conclude that students either attend the lectures and rely mainly on the material offered in class or prefer online material to attending the classes.

5. Conclusions and Recommendations

In this work, we conducted a study involving 224 students in the computer science department enrolled in an introductory Python programming course. The goal of the study is to identify the challenges students face and explore the materials they use and prefer while learning programming. The results provide different challenges that the students face such as understanding the problem, conceiving the solution idea, and using the integrated development environment. We also found that students do not attend private courses outside the university. The most preferred material that the students use is examples and assignment solutions. However, the least helpful method was the course's lecture notes.

Based on the results presented in this study, more attention should be given to topics that students find more challenging to understand. The course content should emphasise how to divide the code into functions and how to use them. This can be achieved by introducing functions early in the course. In addition, a complete revision of the course lecture notes is recommended, considering content based on examples and assignments.

The research's main contribution is to improve the quality of teaching and the content of introductory programming courses. We suggest, as future work, follow-up research to explore the effects of the improvements outlined in this work on the difficulties students face and the materials they prefer for introductory programming courses.

6. References

[1] S. Derus and A. M. Ali, "Difficulties in learning programming: Views of students," in *1st International Conference on Current Issues in Education (ICCIE 2012)*, INDONESIA, 2012.

[2] 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.

[3] A. A.M., "Investigating Difficulties of Learning Computer Programming in Saudi Arabia," *Universal Journal of Educational Research,* vol. 3, no. 9, pp. 567-577, 2015.

[4] S. Singh, "Identifying Learning Challenges faced by Novice/Beginner Computer Programming Students: An Action Research Approach," in *Proceedings of 6th Software Engineering Education Workshop (SEED 2022) co-located with APSEC 2022*, Japan, 2022.

[5] J. Figueiredo and F. J. García-Peñalvo, "Design science research applied to difficulties of teaching and learning initial programming," *Universal Access in the Information Society,* vol. 23, no. 9, p. 1151–1161, 2022.

[6] S. López-Pernas, M. Saqr and O. Viberg, "Putting it all together: Combining learning analytics methods and data sources to understand students' approaches to learning programming," *Sustainability,* vol. 13, no. 9, p. 4825, 2021.

[7] A. Čižmešija, D. Peras and Z. Stapić, "KEY COMPETENCIES AND SKILLS FOR TEACHING AND LEARNING PROGRAMMING: A TERTIARY STUDY," in *16th annual International Conference of Education, Research and Innovation*, Seville, Spain, 2023.

[8] J. Bennedsen and M. E. Caspersen, "Failure rates in introductory programming: 12 years later," *ACM Inroads,* vol. 10, no. 2, p. 30–36, 2019.

[9] M. Mazliah, B. Aslina, I. Rozita, I. Ismassabah, G. M. Ervin, and A. M. N. Noorsidi, "A Preliminary Study of Difficulties in Learning Java Programming for Secondary School," *International Journal of Advanced Trends in Computer Science and Engineering,* vol. 9, no. 1.4, pp. 302-306, 2020.

[10] I. Noman, S. S. Ghazala, F. Ridah and A. Farrukh, "A Study of Difficulties of Students in Learning Programming," *Journal of Education & Social Sciences,* vol. 7, no. 2, pp. 38-46, 2019.

[11] P. H. Tan, C. Y. Ting and S. W. Ling, "Learning difficulties in programming courses: undergraduates' perspective and perception," in *2009 International Conference on Computer Technology and Development (ICCTD 2009)*, Kota Kinabalu, Malaysia, 2009.

[12] C. S. Cheah, "Factors contributing to the difficulties in teaching and learning of computer programming: A literature review," *Contemporary Educational Technology,* vol. 12, no. 2, p. ep272, 2020.

[13] 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.

[14] R. A. M. San Ahmed, M. Sardasht, R. Mahmood, R. M. Nabi and D. L. Hussein, "The impact of teaching materials on learning computer programming languages in Kurdistan region universities and institutes," *Kurdistan Journal of Applied Research,* vol. 3, no. 1, pp. 27-33, 2018.

[15] N. F. M. Noor, A. Saad, A. B. Ibrahim and N. M. Noor, "THE ACCEPTANCE OF AN EDUCATIONAL INTEGRATED DEVELOPMENT ENVIRONMENT TO LEARN PROGRAMMING FUNDAMENTALS," *Information Technologies and Learning Tools,* vol. 93, no. 1, pp. 135-151, 2023.