

Identifying Learning Challenges faced by Novice/Beginner Computer Programming Students: An Action Research Approach

Sarita Singh¹

¹ College of Professional Studies, Northeastern University, USA

Abstract

Educators globally recognize the challenges and complexity of teaching introductory programming courses to novice / beginner computer programming students (novice computer programming students with no or very limited knowledge of computer programming) in the first year of their degree programs. The study seeks to identify and understand the various challenges such students face while learning computer programming through the use of action research methodology. In addition, this study identifies various approaches currently being undertaken at universities and institutes of higher education world-wide to resolve such issues through systematic literature review of peer reviewed journals. The study reflects that problems related to learning computer programming for the first time continue to persist and provides recommendations for future study and action.

Keywords

Action Research, learning challenges, computer programming, novice programmers

1. Introduction

Teaching introductory programming courses is complex. Educators worldwide have recognized the challenges and considerable work has been done in identifying various approaches to teaching computer programming which reflect a blend of content and pedagogy issues [1]. Despite much work being done in these areas, the problems continue to persist [2]. Introductory programming courses at the universities and institutes of higher education equip the students with the necessary skills, knowledge, and expertise to build effective software solutions. With jobs for software developers in high demand due to the rapid advances in technology and increased use of automation across all industries, it is critical for universities and institutes of higher education to retain and produce graduates who are skilled in the field of software development. Evidence of existence of problem related to challenges faced by novice programmers can also be found in literature and there are many examples of extensive studies being conducted in this area [3, 4].

Learning to program is generally considered challenging as it requires cognitive abstraction skills, logical-mathematical skills, and the ability to solve problems using algorithms [2]. In addition, students

suffer from a lack of individual/personal attention on account of large classes and student groups with varying levels of prior programming knowledge. This makes it difficult to design the instruction to be beneficial for everyone which leads to high failure rates causing high dropout rates for computer programming modules [2, 5]. With rapid development in the use of technology in daily life, there is strong demand for skilled computer scientists with strong programming skills [1], and hence it is important to provide students with a strong foundation in computer programming.

My action research project involves identifying the various challenges students face in an introductory programming class at their university through various fieldwork activities such as surveys, interviews and a comprehensive literature review of peer reviewed journals including document analysis.

The action research project is guided by the following primary research questions:

RQ1: What are the learning challenges faced by higher education students without prior computer programming experience who have enrolled in an introductory computer programming course?

Proceedings of 6th Software Engineering Education Workshop (SEED 2022) co-located with APSEC 2022, 06-Dec, 2022, Japan (Virtual)
EMAIL: sarita_s_singh@yahoo.com (S. Singh)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

RQ2: What challenges do instructors perceive while teaching computer programming to such higher education students in an introductory computer programming course?

The survey was conducted with participants pursuing their Bachelor in Computer Science degree program and were students of an introductory programming class. The interview was conducted via Zoom with a faculty member experienced in teaching introductory computer programming courses to students with diverse prior programming background. For the purpose of document analysis, feedback reviews from a popular introductory programming course from a well know online learning platform were analysed.

Teachers of computer science are challenged to help learners develop their understanding of programming. A good understanding of the factors that contribute to the difficulties will help educators in providing appropriate strategies and tools to address these difficulties and help them to better assist their students in their learning journey [6].

The remainder of this paper is organized into four sections describing the research methodology, analysis and key findings followed by conclusion and recommendations for future work.

2. Methodology

This study has been carried out using action research methodology. Action research is grounded in a constructivist paradigm [8]. Action research assumes that all knowledge is constructed socially and is grounded in experiences and perspectives of the people. Action research methodology is also suited for studies based on critical and pragmatic paradigms [9]. Action research uses qualitative techniques to enable participants to get a deeper understanding of ‘how’ and ‘why’ things happen and allow creation of rich information. It allows participants to produce a ‘practical’ solution to the problem through continuous cycles of observation, reflection and action using the ‘Look-Think-Act’ framework. Action research can also use experimental or quasi-experimental research procedures to provide information (i.e., explain what is happening) that can support solutions for the research problem [8, 10].

Action research is an appropriate choice for this study as it focuses on bringing a ‘change’ and allows one to explore reasons for challenges specific to a particular course through active engagement and collaboration with various stakeholders/participants. Data collection through research methods like interviews and surveys help to provide deeper

insights and identify solutions to help resolve issues [8, 11].

2.1 Interview

The interview was conducted using Zoom with a qualified Python (programming language) instructor having several years of teaching experience at major universities in Singapore. The instructor is experienced in teaching the programming languages to undergraduate and postgraduate students, with and without prior programming background.

2.2 Survey

A 15-question survey using a 6-point Likert scale was used to understand the views and opinions of students (with a focus on novice programmers) with regards to the various typical topics studied and related experiences under any introductory programming course. Face validity testing of the survey was conducted with the help of two colleagues at work [12]. (Refer to Appendix A for the survey questions).

Reviewer One was familiar with the fundamentals of programming language and focused on the content and the extent to which the survey items capture the construct whereas Reviewer Two focused on the extent to which the survey items made logical sense, appear confusing or leading, etc. The survey was designed for a group of full-time students (novice/beginner programmers) studying a Python programming course. The course was a 5 week-long program consisting of 4 lessons of 3 hours duration per week (12 hours per week x 5 weeks = 60 hours). The survey was administered at the start of the fifth week when students were working on a project assignment which tested for all the topics covered until then (around 85% of the syllabus). This was an appropriate time to test the areas they found to be difficult as they had to work on the project individually. The survey was designed to challenge students in applying all the knowledge and skills that they had learnt until then.

The survey was further tested for Internal Consistency Reliability by computing Cronbach’s Alpha to measure how well the survey items measure the same construct or whether there is an agreement between the various items designed to measure the same attribute / parameter. Computation of Cronbach’s Alpha resulted in a value of 0.8324144. Since test values of 0.6 or higher indicate good internal consistency, the designed survey instrument was considered acceptable for the analysis.

The survey was created using Google Forms and was shared electronically with the students. 25 responses were recorded. Three incomplete forms were deleted because they had missing data.

2.3 Document Analysis

The document reviewed for this project was the Learner Reviews & Feedback page for a popular introductory computer programming course that is conducted online via an online learning platform.

This course is part of a popular Bachelor of Computer Science program offered by a well-known international university. This MOOC (Massive Open Online Course) provides students with the foundational skill set required to write computer programs. The course introduces the basics of writing and executing computer programs using the JavaScript language and the corresponding library.

The course is delivered by 2 highly experienced and skilled instructors. The course has a rating of 4.1/5 and has a total of 227 reviews. The document for review was selected at random from other such similar course reviews and the first 75 reviews were coded for this project.

2.4 Literature Review

In addition to the above activities, a comprehensive evaluation of various theoretical and empirical work done in peer reviewed journals with regards to teaching and learning computer programming was also carried out.

3. Data Analysis and Key Findings

Exploring the data: A preliminary exploratory analysis was done to get a general sense of the data obtained through interviews and document analysis by annotating the data, and memo-ing the ideas. For this purpose, NVivo was used [13, 14]. Fig. 1 below outlines the visual model of the coding process. The process of coding consisted of identifying text segments and assigning a code that accurately describes the meaning of the text segment.

Visual Model of the CodingProcess

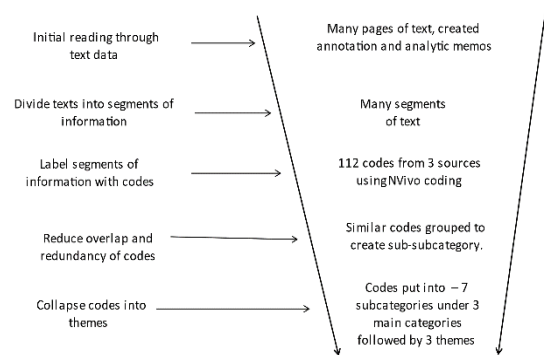


Figure 1: Visual model of the coding process

3.1 Interview

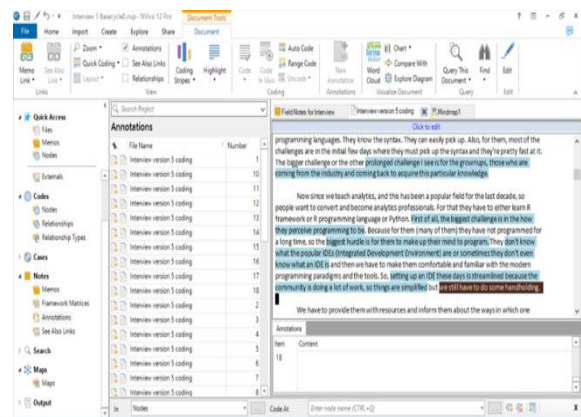


Figure 2: Screenshot of Annotations of Interview transcript

The interview was mainly guided by research question RQ2. The instructor in his interview reaffirms that novice programmers struggle with computer programming courses. He also stated that some of the novice programmers in his class “*don't even know what an IDE is*”. He says, “*So the biggest hurdle is for them to make up their mind to program.*” ‘Change in mindset’ approach is recommended by reviewers of the introductory programming course (document analysis) too. Some of the challenges identified by the instructor were, “*they experience difficulty with being able to use a particular scripting or programming language*” and they have to be “*made comfortable with the syntax*”. Fig 2 above presents the screenshot of the annotations of the interview transcript.

3.2 Survey

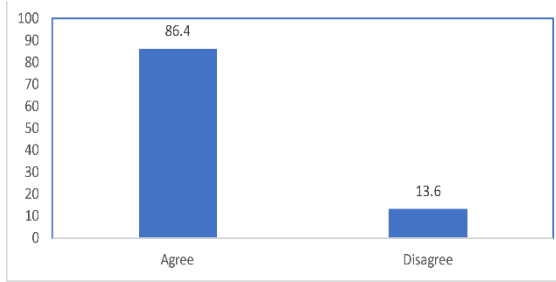


Figure 3: Responses to the question “Novice students find computer programming challenging.”

The survey design was guided by the research question RQ1. Fig. 3 above outlines that 86.4% of the respondents agreed the novice students find computer programming challenging. As seen in Fig.4 and Fig. 5 below, 100% of the respondents agreed that novice/beginner programmers need more support and guidance than those students who have prior programming knowledge.

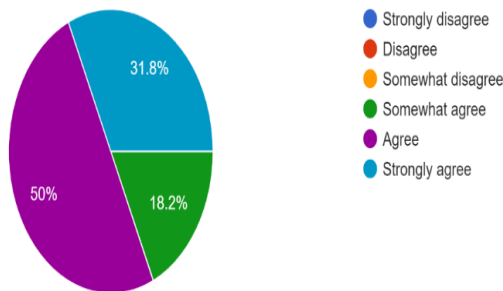


Figure 4: Responses to the question “What percentage of respondents believe that novice/beginner students of computer programming need more support and guidance as compared to those who have prior programming language?”

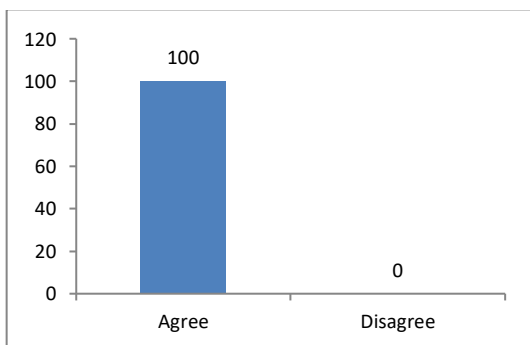


Figure 5: Responses to the question “Novice/beginner programmers need more support and guidance than those students who have prior programming knowledge.”

3.3 Document Analysis

For the document analysis, annotations option was used to get a set of preliminary words or phrases that would help later with the formal coding to determine an appropriate code [13].

<https://www.coursera.org/learn/introduction-to-computer-programming/reviews?page=3>

Annotations

- 1 provide more details
- 2 error in course organization
- 3 not a course for beginners
- 4 lack of mentorship and advise
- 5 no timely respose and feedback for clarification
- 6 download links not working, not beginner freindly course
- 7 poorly designed course
- 8 frustrating course for beginners
- 9 issues with software and instruction
- 10 poor support for windows users
- 11 software not available for download
- 12 recommended software not available
- 13 outdated software
- 14 terribel experience
- 15 unable to download software
- 16 resources not updated
- 17 hands on coding was helpful
- 18 helpful instructor

Figure 6: Example of a list of annotations used

The analysis of the review comments helped to supply an understanding of the challenges that students face in an introductory programming class. The positive reviews helped in naming some of the best practices that could be incorporated in an introductory programming class to help provide a better learning experience to the students. A student in her feedback for the online introductory programming course says, “*It’s impossible to do assignments without a proper explanation, not everyone has already had back experience with JS and since it is an ‘introduction course’ I was expecting more details to how to do assignments*”. Another student says, “*As a person who has no programming background, I was about to quit...It was a stressful experience*”.

Another online student shares similar sentiments, “*This course will make any beginner very frustrated to point of considering quitting and losing excitement about learning to code.*” While we may think that issues may be with the course itself, the course does have positive reviews as well. An online student in her feedback says, “*Highly recommend this course if you have never done anything programming related*”.

3.4 Literature Review

The literature was reviewed through the lens of the intended action research project to identify challenges experienced by novice programmers and how to best support them.

Teaching and learning computer programming is a challenge. High failure and dropout rates are common in introductory programming classes [15, 16, 17, 18, 19]. These are the common introductory statements across most papers written on teaching/learning computer programming related issues. Some of the common reasons identified for various challenges are lack of knowledge, problem solving skills amongst students, large classrooms with high student to teacher ratio resulting in poor student engagement and quality of interaction with students, lack of appropriate learning environment, use of traditional methods that fail to motivate students, complex language syntaxes, abstract concepts that do not have direct analogies in real life, debugging programs, etc. [15, 16, 18, 20, 21, 22].

Many of the students' learning challenges are also connected with the students' prior knowledge [23]. Knowledge of programming can be categorized into three types: syntactic knowledge, conceptual knowledge, and strategic knowledge [7]. Syntactic knowledge refers to the knowledge related to the features of the language such as basic rules and facts, for example, a semicolon is required to end an instruction in Java. Conceptual knowledge refers to the knowledge related to how the various programming constructs work and what is the effect of the execution of these constructs inside the computer. An example of programming construct is a repetitive construct such as a loop. Strategic knowledge refers to the application of conceptual and syntactic knowledge of programming to solve real world problems. This includes planning, coding, and debugging problems, in addition to the knowledge of syntax to develop a working solution [24].

Some of the common errors with regards to syntax is mismatch of parentheses brackets, missing semi colons, failing to declare variables, use of appropriate operators etc. Novice programmers do not understand the concept of variables or that ordering of statements impacts assignment of values to variables [25].

Some of the other factors that contribute to the difficulty experienced by students include cognitive load task complexity, interference of student's understanding of natural language with the meanings of the programming instructions. Novice programmers also face challenges identifying differences between concepts of variable

assignments in programming versus algebraic expressions in math [1, 2, 7].

Studies have also shown that use of digital technologies such as artificial intelligence, machine learning, gamification, visualization tools, augmented and virtual reality etc have the potential to increase motivation amongst students by providing increased visualization, communication, interaction, understanding of complex topics, instant feedback on answers, 24/7 availability, including providing responses for repetitive or frequently asked questions through personalized learning support etc [26, 27, 28, 29]. This can be a solution worth exploring to help students struggling with computer programming classes.

4. Discussions & Reflections

Based on the course document, survey, and interview analysis, it was noted that students do need additional support in terms of additional practice, option for self-paced learning, timely and personalized feedback and appropriate course design and learning environment.

A first course in programming is often a major stumbling block [30]. Programming is a cognitive skill that is complex and novice students encounter difficulties when trying to translate a problem description into a series of solution steps to solve the problem [31].

Other common challenges identified by the instructor were, "*Students have difficulty in understanding how a loop works or understanding adding like structures nested data structures and faculties have to put in lot of effort in explaining these concepts to them.*"

A student from the online course echoes the same in her feedback on loops, "*I thought it would have been good to introduce more content and exercises on loops as this is very fundamental in computer programming. Perhaps a fourth week could have been dedicated to loops*" whereas another says, "*Some basic concepts (e.g., variable) and data structures could be described in more details*".

Students also experienced challenges with the software installation. Many of the open-source software are available only for Windows operating systems and students are increasingly bringing MacBooks to class. Most students do not even know how to use their MacBook well. As a student mentions in his online course feedback, "*Very frustrating coding assignment, with poor support for Windows users and overall tasks.*"

A major theme that emerged from the data analysis is that novice programmers do need additional, personalized, and timely support. A

student in her feedback says, *“For those who never coded before, I would suggest searching some more additional materials on the internet to assist you (at least for me this helped).”* The instructor in his interview also recommended that it is important for the instructors *“to make them comfortable and familiar with the modern programming paradigms and the tools.”* He also states, *“We have to provide them with resources and inform them about the ways in which one installs a particular programming language and the required compilers for the same or sometimes you have to provide the videos.”* Many students from the online course shared about the need for additional support as well. Some of the comments identified from the online course with regards to need for additional support are, *“hard to get help”, “Instructions are not detailed”, “Next to no support in the discussion boards”, “I needed help installing their recommended software, and no one got back to me. I ended up, after much frustration just quitting this class”, “It’s just when you’re truly stuck you can feel a little helpless.”*

Use of whiteboards can also be used to help students visualize abstract concepts. The instructor in his interview states that, *“But then it becomes really difficult for many of the students to understand what the cuboidal form of data structure is, how do I even understand this, so drawing really in my experience helps.”*

Students too seem to appreciate lessons that are interactive and help them to visualize concepts. A student from the online course comments, *“Very results-oriented introduction to programming, focusing on drawing shapes and manipulating them using code.”* Similarly, another says in the feedback, *“I can’t be grateful enough for making an interactive approach to programming”* whereas yet another says, *“Very simple and clear illustration and explanation.”* Use of visualization techniques is recommended by other studies too [20, 19].

It was also noted that not all students work at the same pace. Some take additional time and generally ask questions about a topic much later after it may have already been discussed at length in the classroom. Hence, it is important for some form of support to be available to them as and when they have queries to provide them with a smoother and less stressful learning experience. Also, some students may be too shy to ask questions in large classrooms as was mentioned by the instructor in his interview .

Another theme that emerged from the various field activities is that it is important that the entire course (end to end) be designed well to meet the needs of the novice students. In his interview, the instructor explained at length the need to design and

deliver the course in a manner that a student does not feel too overwhelmed. Missing instructions, outdated software, among others, can add to the learning challenges as noted during class observation as well as from the online course feedback such as, *“too difficult to navigate the menus to find necessary materials”, “most awfully ordered course”, “Too much work to do in order to complete one assignment.”* Positive comments such as *“A positive thing that made me enjoy the course was the methodology applied throughout the classes. You will have a video lecture, and a quiz or exercise after it”*, reinforces the importance of appropriate course design.

5. Future work

The purpose of this action research study is to understand the challenges faced by undergraduate students (novice computer programmers with no or very limited knowledge of computer programming) while learning computer programming. The findings above do help to answer the research question which focused on identifying learning challenges faced by higher education students without prior computer programming experience enrolled in an introductory computer programming course. To begin with, it helped to reaffirm that students with non-programming background do encounter some challenges to learn computer programming and there are cases of students wanting to drop out of the courses as noted in some the review comments as well as in literature [15, 16]. Since the survey design was not very detailed / extensive, it was not possible to get a comprehensive list of challenges. The identification of specific topics/areas of challenge should serve as a guideline for further research to explore these topics in detail and how solutions can be implemented to better support students’ learning.

Some recommended solutions based on the data analysis done for this study to overcome learning challenges are: increased interaction with students, using visualization tools, reduce negative feelings, increase student motivation, engagement and self-efficacy, improve class management, provide immediate continuous feedback and evaluation of learning to students, have appropriate course design & learning environment and use intelligent tutoring systems [21, 15, 16, 20, 22, 18, 19]. It would be worth exploring how digital technologies such as use of artificial intelligence, machine learning, gamification, etc. can help students learn better in classrooms. The focus of this research is to provide a comprehensive solution to all the problems identified.

6. Conclusion

Based on the interview conducted with the faculty member, survey as well as the reviews of the introductory online programming course, it is observed that novice programmers do face challenges while learning computer programming for the first time.

This closely aligns with the findings from other studies identified through literature review as well. Most empirical studies found in literature provide solutions to one or some of the challenges faced by students. Although various solutions and recommendations have been provided by the students, instructor as well as the empirical studies, the findings from this study indicate that a very comprehensive solution addressing all aspects of the various challenges should be designed. Hence it is important to have a complete picture of the multiple challenges faced by students and this can be best achieved by an action research approach.

7. Acknowledgments

Special thanks to all the students who responded to the survey promptly and the instructor for taking time off for the interview from his busy schedule.

8. References

- [1] Koulouri, T., Lauria, S., & Macredie, R. D. (2015). Teaching Introductory Programming: A Quantitative Evaluation of Different Approaches. *ACM transactions on computing education*, 14(4), 1-28. <https://doi.org/10.1145/2662412>
- [2] Malik, S. I., Mathew, R., Al-Nuaimi, R., Al-Sideiri, A., & Coldwell-Neilson, J. (2019). Learning problem solving skills: Comparison of E-learning and M-learning in an introductory programming course. *Education and information technologies*, 24(5), 2779-2796. <https://doi.org/10.1007/s10639-019-09896-1>
- [3] Medeiros, R. P., Ramalho, G. L., & Falcao, T. P. (2019). A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. *IEEE transactions on education*, 62(2), 77-90. <https://doi.org/10.1109/TE.2018.2864133>
- [4] Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2020). A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions. *Computers in human behavior*, 109, 106349. <https://doi.org/10.1016/j.chb.2020.106349>
- [5] Malik, S. I., & Coldwell-Neilson, J. (2016). A model for teaching an introductory programming course using ADRI. *Education and information technologies*, 22(3), 1089-1120. <https://doi.org/10.1007/s10639-016-9474-0>
- [6] Sadler, P. M., Sonnert, G., Coyle, H. P., Smith, N. C., & Miller, J. M., (2013). The influence of teachers' knowledge on student learning in middle school physical science classrooms. *American Educational Research Journal* 50, 5 (2013), 1020–1049. DOI:<http://dx.doi.org/10.3102/0002831213477680>
- [7] Qian, Y., & Lehman, J. (2017). Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. *ACM transactions on computing education*, 18(1), 1-24. <https://doi.org/10.1145/3077618>
- [8] Stringer E. T., & Aragon A. O. (2021). *Action Research*. (Fifth Edition ed.) SAGE Publications, Inc.
- [9] Kivunja, C., & Kuyini, A. B. (2017). Understanding and applying research paradigms in educational contexts. *International Journal of higher education*, 6(5), 26-41.
- [10] MacDonald, C. (2012). Understanding participatory action research: A qualitative research methodology option. *The Canadian Journal of Action Research*, 13(2), 34-50.
- [11] Herr, K. & Anderson, G. L. (2015). *The Action Research Dissertation: A Guide for Students and Faculty* (Second Edition) SAGE Publications, Inc.
- [12] Portnoy, L. (n.d.). Understanding Validity in Survey Instruments
- [13] Creswell, J. W. (2012). *Educational research: planning, conducting, evaluating, quantitative and qualitative research* (Fourth Edition). United State of America: Pearson Education Inc
- [14] Saldaña. (2021). *The coding manual for qualitative researchers* (4E [Fourth edition]). SAGE.
- [15] Bati, T. B., Gelderblom, H., & van Biljon, J. (2014). A blended learning approach for teaching computer programming: design for large classes in Sub-Saharan Africa. *Computer science education*, 24(1), 71-99. <https://doi.org/10.1080/08993408.2014.897850>
- [16] Cardenas-Cobo, J., Puris, A., Novoa-Hernandez, P., Parra-Jimenez, A., Moreno-Leon, J., & Benavides, D. (2021). Using Scratch to Improve Learning Programming in College Students: A Positive Experience from a Non-WEIRD Country. *Electronics (Basel)*, 10(10),

1180.
<https://doi.org/10.3390/electronics10101180>
- [17] Kanaparan, G., Cullen, R., & Mason, D. (2019). Effect of Self-efficacy and Emotional Engagement on Introductory Programming Students. *AJIS. Australasian journal of information systems*, 23. <https://doi.org/10.3127/ajis.v23i0.1825>
- [18] Unal, A., & Topu, F. B. (2021). Effects of teaching a computer programming language via hybrid interface on anxiety, cognitive load level and achievement of high school students. *Education and information technologies*, 26(5), 5291-5309. <https://doi.org/10.1007/s10639-021-10536-w>
- [19] Xinogalos, Satratzemi, M., & Malliarakis, C. (2015). Microworlds, games, animations, mobile apps, puzzle editors and more: What is important for an introductory programming environment? *Education and Information Technologies*, 22(1), 145–176. <https://doi.org/10.1007/s10639-015-9433-1>
- [20] Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2011). Improving teaching and learning of computer programming through the use of the Second Life virtual world: Improving teaching and learning through Second Life. *British journal of educational technology*, 42(4), 624-637. <https://doi.org/10.1111/j.1467-8535.2010.01056.x>
- [21] Altintas, T., Gunes, A., & Sayan, H. (2016). A peer-assisted learning experience in computer programming language learning and developing computer programming skills. *Innovations in education and teaching international*, 53(3), 329-337. <https://doi.org/10.1080/14703297.2014.993418>
- [22] Ling, H.-C., Hsiao, K.-L., & Hsu, W.-C. (2020). Can Students' Computer Programming Learning Motivation and Effectiveness Be Enhanced by Learning Python Language? A Multi-Group Analysis. *Frontiers in psychology*, 11, 600814-600814. <https://doi.org/10.3389/fpsyg.2020.600814>
- [23] DiSessa, A.A., (2014). The construction of causal schemes: Learning mechanisms at the knowledge level. *Cognitive science*, 38(5), 795–850. <https://doi.org/10.1111/cogs.12131>
- [24] Volet, S.E., & McGill, T.J., & Simone E., (1997). A conceptual framework for analyzing students' knowledge of programming. *Journal of Research on Computing in Education* 29, 3 (1997), 276–297. DOI:<http://dx.doi.org/10.1080/08886504.1997.10782199>
- [25] Kölling, M., & McCall, D., & (2014). Meaningful categorisation of novice programmer errors. In 2014 IEEE Frontiers in Education Conference Proceedings (FIE'14). IEEE, 1–8.
- [26] Chen, H.-L., Vicki Widarso, G., & Sutrisno, H. (2020). A ChatBot for Learning Chinese: Learning Achievement and Technology Acceptance. *Journal of educational computing research*, 58(6), 1161-1189. <https://doi.org/10.1177/0735633120929622>
- [27] Vanichvasin, Patchara. (2021). Chatbot Development as a Digital Learning Tool to Increase Students' Research Knowledge. *International Education Studies*. 14. 44. 10.5539/ies.v14n2p44.
- [28] Vázquez-Cano, E., Mengual-Andrés, S., & López-Meneses, E. (2021). Chatbot to improve learning punctuation in Spanish and to enhance open and flexible learning environments. *International Journal of Educational Technology in Higher Education*, 18(1), 1-20. <https://doi.org/10.1186/s41239-021-00269-8>
- [29] Yin, Goh, T.-T., Yang, B., & Xiaobin, Y. (2021). Conversation Technology with Micro-Learning: The Impact of Chatbot-Based Learning on Students' Learning Motivation and Performance. *Journal of Educational Computing Research*, 59(1), 154–177. <https://doi.org/10.1177/0735633120952067>
- [30] Proulx, V. K. (2000). Programming patterns and design patterns in the introductory computer science course. In SIGCSE 2000: Proceedings of the 31st SIGCSE technical symposium on Computer science education, pages 80–84. ACM
- [31] Gerdes, A., Heeren, B., & Jeuring, J. (2009). *Constructing Strategies for Programming - Proceedings of the 1st International Conference on Computer Supported Education (Vol. 1)*.

Appendix A

Please respond to the following based on your experience after four weeks of studying Python programming in this class. It is highly recommended that this survey be attempted by only novice/beginner programmers (students).

1. I consider myself as a novice/beginner (i.e., with no or minimal prior computer programming knowledge) computer programmer.
2. Students learning computer programming for the first time (novice/ beginner programmers) find computer programming challenging.

3. Novice programmers experience difficulty in comprehending (understanding) the requirements through the problem statement of a given assignment.
4. It is challenging for novice programmers to set up and use the Integrated Development Environment (IDE) on their own.
5. Novice programmers find it challenging to debug their computer programs for errors on their own.
6. As a novice programmer, I experienced difficulty in understanding the concept of variables and data types.
7. As a novice programmer, I experienced difficulty in using appropriate syntax while writing my programs.
8. As a novice programmer, I experienced difficulty in visualizing and using appropriate programming constructs such as 'while' and 'do- while' loops.
9. As a novice programmer, I experience difficulty in implementing 'functions/methods' in my program.
10. As a novice programmer, I experience difficulty in designing a program to solve a given task.
11. As a novice programmer, I experienced difficulty in understanding Object-Oriented related concepts.
12. As a novice programmer, I experienced difficulty in defining a class consisting of its attributes and related implementations.
13. Novice programmers need more support and guidance as compared to those students who have prior programming knowledge.