Zamfara International Journal of Education (ZIJE)
The Official Journal of the Faculty of Education
Federal University Gusau, Zamfara State, Nigeria
Volume 3, Number 1, June, 2023
pISSN: (Hard Copy): 2814 – 1377; eISSN (Online): 2814 – 1369
www.zijedufugusau.com

DOI: https://doi.org/10.5281/zenodo.8370788

Python Coding amongst Undergraduate Student Teachers in a Nigeria Post-Secondary Institution: An Exploratory Study

***Bakare, Olusegun Ojo[1]**
oluseguns.bakare@gmail.com

Opara, Chima Martins[1]
chimamartins10@gmail.com

[1]Obafemi Awolowo University, Ile-Ife, Nigeria

Abstract

This study explored coding errors, determined how preservice teachers in Obafemi Awolowo University (OAU) debugged coding error and investigated reasons for the errors. This was with an intention to assess preservice teachers learning and understanding of python programming in OAU. The study adopted exploratory research design with the population of all undergraduate students in Faculty of Education in OAU. Out of the population, 10 preservice teachers were interviewed to understand the nature of the study. The interviews were audio recorded, transcribed and thematically analysed. The study findings revealed four major errors committed in python programming by student teachers in the university. The finding further categorized approaches adopted by the students in debugging python programming errors into seven. Amongst these seven, 'help from friends and internet' were predominant while the least was 'doing it again'. The undergraduate students' teachers advanced six reasons for the errors often faced while learning programming. The paper concluded that undergraduate student teachers confront errors while learning python programming. It was strongly recommended that the pedagogy of computer programming should be taught inhouse to improve programming learning among undergraduate student teachers.

Keywords: Python Programming, Coding Errors, Undergraduate Student Teachers, Exploratory Research, Debugging

Introduction

The term coding and programming can be used interchangeably. San Ahmed, et al., (2018) regarded computer programming as the procedure for "writing, testing and debugging computer programs" using diverse programming languages (P. 27). This act of writing set of codes, testing and debugging could be referred to coding. Research argued that learning coding is complex (San Ahmed et al., 2018). Despite the complexity in coding, learning computer programming is no longer restricted to engineering and pure science students of Obafemi Awolowo University (OAU) only but also involving preservice students in the core science related studies. All students who enrolled to study Mathematics, Physics, Chemistry and Biology education are expected to register and pass the borrowed courses from Computer Department of the University. The programming courses consisted of both theoretical and practical. According to Muslihudin, et al., (2021), the reason for learning programming languages is to develop application software for users. Far above this, observation reveals that the purpose of student teachers learning the acts of coding in this present era is to develop the pedagogy of programming and thinking skills that would be required to teach Computer Studies in Nigerian primary and secondary schools. Since the pedagogy of computer programming might not be taught outside the domain of education faculty, the complexity of learning computer programming asserted by San Ahmed et al. (2018) is further strengthen.

Similarly, a good number of studies (Brenda, et al., 2023; Mwaba, 2022) have shown that there are few qualified teachers teaching Computer Studies in primary and secondary schools. This limitation could be traced to teachers' development in computer programming during their training in the tertiary institutions. Even undergraduate students who are outside education faculty but exposed to computer programming equally faced with coding challenges. For example, in a study conducted by Jegede, et al., (2019) to investigate java programming errors committed amongst computer science and engineering students in Southwestern universities in Nigeria. The study identified five key errors, namely; Invalid Symbol, Mismatch Symbol, Missing Symbol, Inappropriate Naming, and Excessive Symbol. However, Missing symbols (33%) and Invalid symbols (12%) were regarded as the highest and least committed errors respectively. Similarly, in recent research of Jegede, et al., (2023) which categorized the errors committed by engineering students in python programming into five. These errors include; Missing Symbols (30.0%), Invalid symbols (21.1%), Mismatched symbols (25.6%), Inappropriate naming (8.9%) and Excessive symbols (14.4%). The study further revealed that the topmost committed errors fall on Input and Output concepts (35.6%), as well as Loops (25.6%). Additionally, Kazemitabaar, et al., (2023) categories coding errors committed by learners into three; data-types, syntax, and semantics. These reviewed literatures revealed that divers programming errors are being committed by students. The researchers identified the errors. However, little attention was given to how these learners have been overcoming the complexity embedded in coding as they learn programming. Therefore, this current research intended to feel this gap by providing insight to how students teachers have been facing similar errors but debugging to get their codes running. It further provided implications for teaching and learning programming in Nigerian primary and secondary schools. To this end, the study assessed pre-service teachers' learning of python programming in Obafemi Awolowo University, Ile-Ife, Osun State.

The concepts of programming errors have been explained in the literatures. Prominent among the programming errors include Syntax Errors, Logic Errors, Runtime Errors, Semantic Errors, and Null Pointer Exceptions.

1. **Syntax Errors**: These errors occur when the code violates the rules of the programming language (Mase & Nel, 2021). Syntax Errors are committed due to invalid symbols, mismatched symbols, missing symbols, or inappropriate naming.
2. **Logic Errors**: Logic errors occur when the code does not produce the expected output due to flaws in the program's logic. These errors are often difficult to identify and can lead to unexpected behaviour or incorrect results. Ettles, et al., (2018) classified logic errors into three, namely; algorithmic errors, misinterpretation of the problem, and fundamental misconception.
3. **Runtime Errors**: Runtime errors occur during the execution of the program and can cause it to crash or terminate unexpectedly (Garion, et al., 2022). According to Kästner, et al., (2022) the errors are due to undefine or unspecify actions of the programming language in question. These errors can be caused by issues such as division by zero, accessing invalid memory, or using uninitialized variables (Liu et al., 2023).
4. **Semantic Errors**: Semantic errors occur when the grammatical rule of coding is violated or that the code is syntactically correct but does not perform the intended functionality (Silva, 2020). These errors are challenging to identify as they may not result in immediate errors or warnings.
5. **Null Pointer Exceptions**: Null pointer exceptions occur when a program attempts to access or manipulate an object or variable that has a null value. These errors can lead to program crashes and require careful handling of null values. Kästner et al. (2022) categorised dangling pointer accesses, invalid pointer dereferences and arithmetic, null pointer dereferences, misaligned dereferences, buffer overflows as Invalid pointer dereference and manipulation

To avoid these errors, it is essential to follow coding standards and best practices, use an integrated development environment (IDE) with syntax highlighting and error checking, and thoroughly test the code for all possible scenarios.

Objectives of the Study

Three specific objectives guided this study, these are to:

1. Assess the coding errors that preservice teachers in OAU do commit while learning python programming;
2. Determine how the OAU preservice teachers do debug the errors in python programming;
3. Examine the reasons informing the errors committed by the preservice teachers in python programming?

Research Questions

Three research questions were raised from the objective. These are:

1. What coding errors do preservice teachers in OAU commit while learning python programming?
2. How do OAU preservice teachers debug the errors in python programming?
3. Why do the OAU preservice teachers commit the errors in python programming?

Methodology

This paper adopted exploratory research design to elicit in-depth data from seven undergraduate student teachers. The sample size of 10 was randomly drawn from the population consisted of all students in the Department of Science and Technology Education in Faculty of Education, Obafemi Awolowo University, Ile-Ife, Nigeria. According to Creswell and Creswell, (2017) researchers could estimate between 10 and 50 participants as being appropriate depending on type of research and research questions. Based on this assertion the selection of 10 participants could be termed appropriate. This sample size was used to give in depth insight to the study. Similarly, the department was purposively selected since it was the only one offering computer programming in the Faculty of Education. An interview guide was used to elicit data from the participants. The trustworthiness of the interview was obtained. In this study, the transcribed audio recordings (using Notta software) of the interviews were read, coded, and organised into themes (Dhlamini, 2022). To obtain the trustworthiness of the study, the researchers read the transcribed lesson study cycle, lesson observation notes, audio recordings repeatedly to understand what the undergraduate teachers were saying in the course of the interview. The trustworthiness of qualitative research could be established using four strategies, these are: audibility, confirmability, credibility and transferability. The data was Thematic Analysis Approach for the analysis (Creswell & Creswell, 2017). A thematic analysis approach was used to identify patterns and themes in the participants' responses. The analysis process involved several steps, including familiarity of the data, coding of the data, and identification of themes. The themes were then further analysed descriptively to draw meaningful insights and results obtained.

Results

Research Question 1

What coding errors do preservice teachers in OAU commit while learning python programming?

Table 1: Coding errors committed by student teachers while learning python programming

Comments
Spk1: ... but sometimes, I missed some commands like all those commas, and the computer shows some errors but not running.
Spk2: …Using the wrong functions, hmm maybe skipping some step, and not inputting the right commands
Spk3: ...maybe where I am supposed to put dot, where I'm supposed to use some marks and stuffs like that.
Spk6: ...understanding the concept of loops, functions and modules. I also found it difficult to debug my code when I encountered such errors
Spk7: ...I found it difficult to remember the syntax and what I needed to use or do
Spk9: .... from full-stop, commas

From the excerpts in Table 1 above, the student teachers identified six coding errors. Prominent among these errors were; missing commands, syntax errors, use of wrong functions and modules, skipping steps, logics, and looping. Additionally, Figure 1 was used to represent the preservice teachers' comments regarding the coding errors often committed quantitatively.

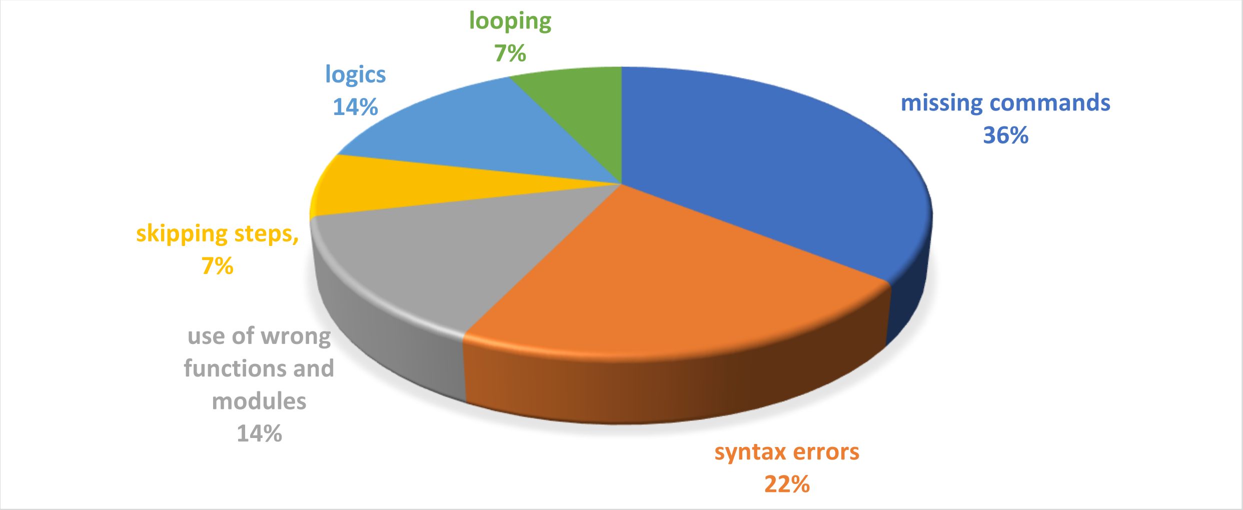

Figure 1: Thematic Analysis of Python Errors Among Preservice Teachers from OAU

From Figure 1, 36% of the preservice teachers in Obafemi Awolowo University, commit missing commands errors, 22% syntax errors, 14% logics and use of wrong function/modules errors, while 7% commit looping and skipping steps. It can be concluded that off all these errors, the students' teachers mostly committed missing command errors, followed by Syntax which often involved issues such as missing or misplaced punctuation, incorrect indentation, and incorrect use of python keywords.

Research Question 2

How do OAU preservice teachers debug the errors in python programming?

With respect to the above research question, the student teachers were interviewed and audio recorded. The excerpts of their comments were illustrated in Table 2 below to show the different ways by which the preservice teachers were able to debug the various errors encountered during their learning of python programming.

Table 2: Debugging strategies adopted by student teachers while learning python programming

	Comments
Speaker 1	*...doing it again and trying to know why it's bringing the errors, … I read through the lower part of the python interface or something, there is a red node that shows that this one is due to syntax, so I see some of those arrows and you can easily say, oh this is where I make a mistake and just fixing the writing, I get good results.*
Speaker 3	*...at least little ideas from my friends and everybody around that had understanding about it and, some people browsed the internet so I was able to get some helpful materials and all of that*
Speaker 4	*… I do like create a file to use to access it and, doing trial and error most times but my friends have been helpful.*
Speaker 5	*...I don't know, probably trial and error sha to see if it works out or not or get assistance from my friends.*
Speaker 6	*...by practicing more, watching online tutorials, and asking for help from my friends, classmate and instructors. I also tried to breakdown the concept into smaller parts and understand them one at a time*
Speaker 7	*...I did everything on my own by practicing more and using online resources like YouTube videos and python documentation*
Speaker 8	*...by creating a personal cheat sheet for functions and libraries that I use often. I tried to understand the logic behind each function and how it can be applied in each scenario*
Speaker 9	*...scanned through my work, and saw where I did not put full-stop, commas and some little things and then corrected it*
Speaker 10	*...I tried as much as possible to read and watch YouTube videos before going for practical*

From the excerpts in Table 2 above, the students identified seven strategies which they used to debug coding in Python. First; *doing it again*, help from friends, help from internet (YouTube), practicing more, creating file/sheet to access it better, trial and error and understanding the logic and breaking it down in order to debug it properly. The strategies used are represented in Figure 2.

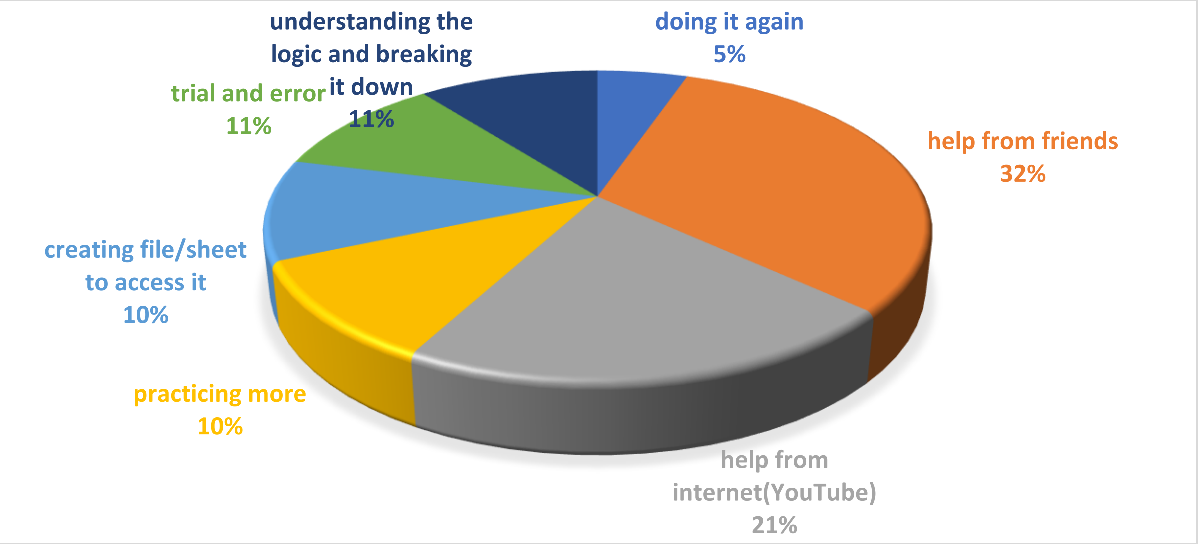

Figure 2: Python Programming Debugging Methods

From Figure 2, the result showed seven strategies that the student teachers adopted to solve errors while coding in python. These strategies are help from friends (32%), help from internet (YouTube) (21%), understanding the logic and breaking it down (11%), trial and error (11%), creating file/sheet to access it (10%), practicing more (10%), and doing it again (5%). The most commonly mentioned strategy was peer collaboration, seeking help from friends that are good programming, followed was using online resources such as YouTube videos and other python documentations to find solutions to their coding errors and also to have more knowledge about python at large. Some preservice teachers also went ahead to practice more on their own as a strategy to debugging coding errors. While trying to do it over again was least of the strategies used in debugging coding errors.

Research Question 3

Why do OAU preservice teachers commit the errors in python programming?

With respect to the above research question, the student teachers were interviewed and audio recorded. The excerpts of their comments were illustrated in Table 3 to show reasons the categories of errors faced while learning of python programming.

Table 3: Reasons informing python programming errors committed by student teachers

	Comments
Speaker 1	*…some instructors or lecturers who make it even more tough. It's something you're learning, it should be interesting but then they make it look very hard and you know, it should be fun and interesting ,… if students can get their own personal laptop which they can be using to learn on a practical note, … ehn not so good environment to learn*
Speaker 2	*…like more practical should be included, like more of training, physical training. Students been invited to the lab to actually practice what they are been taught and to put their manuals into use.*
Speaker 3	*…basically, it's a tip of the iceberg that we are doing here. They are not too detailed on it, … institution like OAU should employ more programming experts not only on python programming. They should encourage it because it's kind of something people are really into now*
Speaker 4	*… organize maybe physical class and not this kind of online of a thing where we can have a one-on-one interactions with lecturers and the lecturers would be able to help us out or put us through in learning it better*
Speaker 5	*… probably give us more examples, and make it more explanatory so that they get to know what we are doing if we are doing the right thing*
Speaker 6	*… opportunities for hands on learning and real-world projects would be helpful. It would also be helpful to have one-on-one support and resources for debugging and error handling*
Speaker 7	*… materials and equipment available to students and also, a one-on-one encountered should be okay and also, …hands-on practice and more opportunities to collaborate with other students*
Speaker 8	*… enough facilities and instructors*
Speaker 9	*… more instructors to student ratio because we have not so many lecturers taking the course and also, make it a form of class tutorial*
Speaker 10	*… intelligent instructors to help facilitate learning process*

From the excerpts in Table 3 above, the preservice teachers identified six reasons why they committed the errors while learning python programming in OAU. The reasons include; Bad learning environment, limited practical's, limited instructors/experts in the field, limited facilities for learning, no detailed explanations and little or no examples to backup what they have learnt.

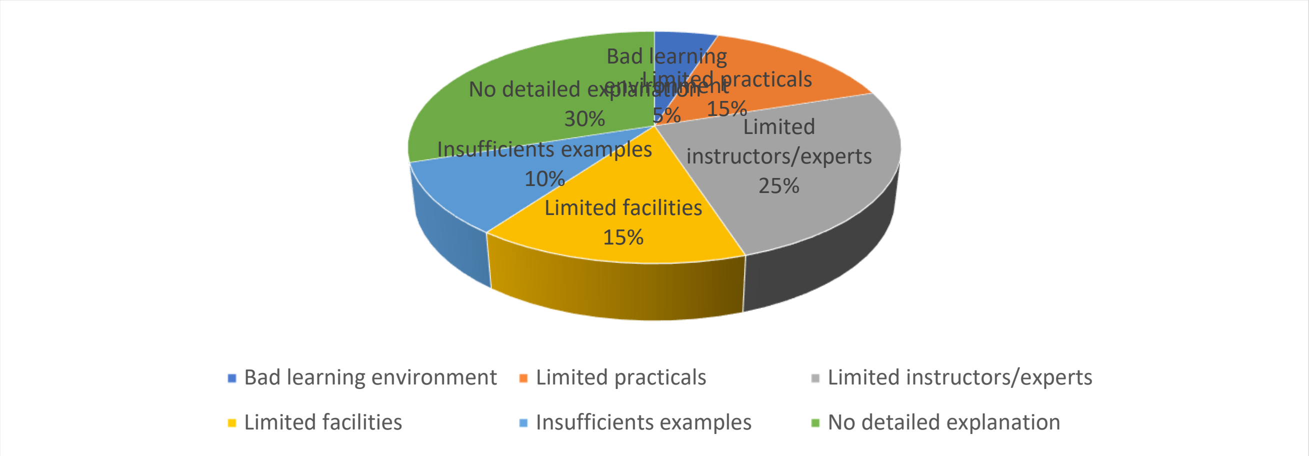

Figure 3: Reasons for Python Programming Errors Amongst OAU Students

The analysis of the interview data was categorized into six reasons why the student teachers committed the coding errors while learning python programming. One common reason was lack of detailed explanation on the part of the instructors, thinking that students already know what python and coding is all about. Preservice teachers also gave reasons that there were limited instructors and experts to explain in details what they should know about python programming. Also, another major reason OAU preservice teachers commit the errors in python programming is due to the fact that there are limited facilities which can go round once in the learning process as some students might miss out on some important information which the instructor would have passed across to the first batch as there is no uniformity. A few preservice teachers also said insufficient examples, limited practical and bad learning environment played a major factor as to the reasons they committed coding errors while learning python programming as they were unable to explore more due to lack of proper understanding.

Discussion of Findings

The study findings indicate six coding errors. These errors were 36% missing commands errors, 22% syntax errors, 14% logics and use of wrong function/modules errors, while 7% commit looping and skipping steps. The most prevalent of these errors was missing commands errors followed by syntax errors. This finding is consistent with previous research of Jegede et al. (2019) who found missing symbols and syntax toping errors committed by novice programmer. Similarly, Kazemitabaar et al. (2023) and Jegede et al. (2023) identified syntax errors as one of the errors often committed by learners. The prevalence of syntax error among other errors committed by students learning programming suggested that undergraduate student teachers may need more programming courses taught within their faculties, practice more, organised programming tutorials as well as guidance in understanding and applying the correct syntax in their code.

The findings also highlighted seven debugging strategies adopted by the undergraduate student teachers. These strategies are, help from friends (32%), help from internet (YouTube) (21%), understanding the logic and breaking it down (11%), trial and error (11%), creating file/sheet to access it (10%), practicing more (10%), and doing it again (5%). In previous study, use of YouTube to find help for codes and material, shared were suggested as a strategy to debugging (Weisberg, Schinazi, Ferrario, & Newcombe, 2023).

The study furthermore revealed six reasons why the student teachers committed the errors while learning coding. The reasons include: no detailed explanation (30%), limited instructors/experts (25%), limited practical (15%), limited facilities (15%), insufficient examples (10%) and bad learning environment (5%). The most mention challenge was no detailed explanation (30%), followed by limited instructors/experts (25%) while bad learning environment (5%) was less mentioned. The finding is similar to that of Jegede et al. (2023) who pointed out the need for efficient programming practical classes organised to build students coding skills amongst university undergraduates. Providing students with sufficient practice exercises and detailed explanations and opportunities for peer collaboration can mitigate these challenges and improve their overall coding skills.

Conclusion

The findings revealed that student teachers faced coding bugs like missing commands errors, syntax errors, logics and use of wrong function/modules errors, as well as looping and skipping of steps errors. The student teachers also strategies various approaches to debug the errors confronting them while coding in python programming language. The findings also provided valuable insights to reasons for facing these challenges. Based on all these, this paper provided suggestions to enhance the teaching and learning of programming amongst preservice teachers.

Recommendations

The paper recommended that:

1. Pedagogy of programming languages should be taught in house the teachers training faculties across the country to improve preservice teachers' skills in programming languages;
2. Practical tutorial should be organised for undergraduate student teachers;
3. Access to internet facilities should be provided so that students learning programming can find help using YouTube and other resources

References

Brenda, T. T., Odili, J. N., & Osadebe, P. U. (2023). Assessment of the level of achievement of students in computer studies in senior secondary schools in Bayelsa State, Nigeria. *International Journal of Innovative Education Research, 11*(1), 58-65. https://seahipaj.org/journals-ci/mar-2023/IJIER/full/IJIER-M-6-2023.pdf

Creswell, J. W., & Creswell, J. D. (2017). Research design: Qualitative, quantitative, and mixed methods approaches: Sage publications. http://www.ceil-conicet.gov.ar/wp-content/uploads/2015/10/Creswell-Cap-10.pdf

Dhlamini, S. S. (2022). Exploring the role of lesson study on educators′ pedagogical content knowledge in teaching mathematics. https://openscholar.dut.ac.za/bitstream/10321/4542/3/Dhlamini_S_2022_Redacted.pdf

Ettles, A., Luxton-Reilly, A., & Denny, P. (2018). Common logic errors made by novice programmers. In ACE 2018: *20th Australasian Computing Education Conference*. https://doi.org/10.1145/3160489.3160493

Garion, C., Hattenberger, G., Pollien, B., Roux, P., & Thirioux, X. (2022). A gentle introduction to C code verification using the Frama-C platform. ISAE-SUPAERO; ONERA--The French Aerospace Lab; ENAC. . (pp. 1-18) https://hal.science/hal-03625208/document

Jegede, P. O., Olajubu, E. A., Bakare, O. O., Elesemoyo, I. O., & Owolabi, J. (2023). Analysis of Syntactic Errors of Novice Python Programmers in a Nigeria University. In K. Arai (Ed.) *Intelligent Computing: Proceedings of the 2023 Computing Conference*. (pp. 285–295). Springer. https://doi.org/10.1007/978-3-031-37963-5

Jegede, P. O., Olajubu, E. A., Ejidokun, A. O., & Elesemoyo, I. O. (2019). Concept–based Analysis of Java Programming Errors among Low, Average and High Achieving Novice Programmers. *Journal of Information Technology Education. Innovations in Practice, 18*, 49-59. https://doi.org/10.28945/4322

Kästner, D., Mauborgne, L., Wilhelm, S., Mallon, C., & Ferdinand, C. (2022). Static data and control coupling analysis. In: *11th Embedded Real Time Systems European Congress (ERTS2022)*. https://hal.science/hal-03694546/document

Kazemitabaar, M., Chyhir, V., Weintrop, D., & Grossman, T. (2023). Scaffolding Progress: How Structured Editors Shape Novice Errors When Transitioning from Blocks to Text. In: *The 54th ACM Technical Symposium on Computer Science Education*. 1. https://doi.org/10.1145/3545945.3569723

Liu, Y., Le-Cong, T., Widyasari, R., Tantithamthavorn, C., Li, L., Le, X.-B. D., & Lo, D. (2023). Refining ChatGPT-generated code: Characterizing and mitigating code quality issues. ACM. 1(1) https://arxiv.org/pdf/2307.12596.pdf

Mase, M. B., & Nel, L. (2021). Common Code Writing Errors Made by Novice Programmers: Implications for the Teaching of Introductory Programming. In: *The Annual Conference of the Southern African Computer Lecturers' Association*. 102–117. https://link.springer.com/chapter/10.1007/978-3-030-95003-3_7

Muslihudin, M., Purnama, R., & Lestari, U. (2021). Application design for identification of car garbage prone points for pringsewu regional government cleanliness based on mobile web. *International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455-5290, 1*(01), 22-36. http://journal.hmjournals.com/index.php/IJITC/article/download/1036/1190

Mwaba, B. (2022). An investigation of academic performance of grade 9 learners in computer studies: a case study of Chamunda secondary school in Masaiti district, Zambia. The University of Zambia.

http://41.63.25.23/bitstream/handle/123456789/7661/FINAL_PRINT_BEATRICE_BEATRICE07_1_2019.pdf?sequence=1&isAllowed=y

San Ahmed, R. A. M., Sardasht, M., Mahmood, R., Nabi, R. M., & Hussein, D. L. (2018). The impact of teaching materials on learning computer programming languages in Kurdistan region universities and institutes. *Kurdistan Journal of Applied Research, 3*(1), 27-33. http://kjar.spu.edu.iq/index.php/kjar/article/download/177/140

Silva, R. C. d. (2020). Enhancing textual explanations for Java methods with variable role knowledge. https://repositorio.iscte-iul.pt/bitstream/10071/22092/4/master_ricardo_cardoso_silva.pdf

Weisberg, S. M., Schinazi, V. R., Ferrario, A., & Newcombe, N. S. (2023). Evaluating the effects of a programming error on a virtual environment measure of spatial navigation behavior. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 49*(4), 575. https://scholarshare.temple.edu/bitstream/handle/20.500.12613/8229/Newcombe-Preprint-2022.pdf?sequence=1.