

Student and instructor perceptions of C as a first programming language in higher education in the Tanzanian context

Kaanael Simon Mbise¹
Anna Mark Kayanda²

¹kaanael.mbise@iaa.ac.tz

²anakayanda@gmail.com

¹Institute of Accountancy Arusha, ²College of Business Education, ^{1,2}Tanzania

<https://doi.org/10.51867/ajernet.7.2.91>

ABSTRACT

The selection of appropriate introductory programming languages for Information Technology students has been a topic of continuous concern in computing education research. In this study, we explore both students' and lecturers' perceptions of using C as a first programming language and its contribution to the perceived transferability of programming knowledge to other programming languages within Tanzanian higher learning institutions. The study targeted all 158 second- and third-year students of the Bachelor of Information Technology program at the Institute of Accountancy Arusha and the College of Business Education, as they have completed an introductory programming course using the C programming language. Data was collected from 96 second- and third-year students as well as 42 lecturers from Tanzanian higher learning institutions that teach C programming. A quantitative research design was employed, utilizing structured questionnaires for data collection from both students and lecturers. Descriptive analysis and separate regression models were used, given the importance of examining perceived differences in students' (learning) and lecturers' (teaching) perspectives regarding C programming. The results revealed that students perceived C as a fairly appropriate choice for a first programming language. Student-based regression analysis indicated that the perceived ease of learning C programming positively predicted the transferability of programming knowledge to other languages. Similarly, lecturer-based regression revealed a strong correlation between the perceived ease of teaching C programming and its transferability to other programming languages. It is important to note, however, that these models are based on perceptions rather than actual learning outcomes. Nonetheless, the study provides empirical evidence in the Tanzanian context, which has been an overlooked area in computing education research. Future research focusing on actual learning outcomes and comparative studies of different programming languages will be crucial for reinforcing the findings of this study. In terms of policy and practice, the findings underscore the need for curriculum designers and higher education institutions in Tanzania to critically reassess the selection of introductory programming languages with respect to both the student learning experience and the effectiveness of teaching. Policymakers are encouraged to promote curriculum reforms informed by robust evidence from comparative studies of programming languages and cross-platform skill transferability.

Keywords: C Programming Language, First Programming Language, Higher Education Students, Programming Language, Tanzania

I. INTRODUCTION

A computer system is composed of two essential parts, namely hardware and software. The hardware consists of physical devices that make up a computer. The software includes all programs that run on a computer (Szeliski, 2022). Software is very important as it controls what computers and associated devices can do (Edafe, 2020). Computer software makes computers perform specific tasks for users, developers and system administrators. A computer can do anything that is instructed by the user if it has the appropriate software to do the task (Szeliski, 2022). Computer professionals called programmers or software developers design and write computer software using programming languages (Deitel & Deitel, 2022).

There are hundreds of programming languages that can be used to write computer software. Computers execute instructions written in programming languages directly or require intermediate programs for conversion or translation (Dmitrović, 2021). Programming languages may be divided into three general types, namely machine languages, assembly languages, and high-level languages (Deitel & Deitel, 2022). Computers understand programs written in machine languages while programs written in assembly and high-level languages require assemblers and compilers respectively. Programmers or software developers choose a particular programming language to use for developing software depending on the application's requirements (Perry & Miller, 2017). Examples of programming languages are C, C++, C#, Java, PHP, JavaScript, Ruby, and Python.

Dennis Ritchie invented the C programming language in 1972 at Bell Laboratories (Kernighan & Ritchie, 2017). C evolved from B programming language. The initial use of C programming language was the development of the UNIX

operating system. Since its invention, the C programming language has become a popular programming language for the development of operating and application software (Kochan, 2014; Seacord, 2020). The C language provides strong features such as simplicity, machine portability, mid-level language, efficient memory management, structured programming language, fast speed execution, rich C standard library, and extensibility (Deitel & Deitel, 2022; Gonzalez, 2021). Because of the C language’s fast speed, it is commonly used for the development of computer systems that require performance such as communication systems, real-time systems, embedded systems, and operating systems (Szuhay, 2022).

A debate on the choice of the ideal first programming language for teaching students in information technology (IT) related fields the basics of programming has attracted attention for years (Prokop et al., 2019). Several researchers have conducted studies on how the choice of the first programming language course affects the learning of other subsequent programming languages for higher education students. For example, Chen et al. (2019) conducted a comparison study of the choice of the first programming language in university students between graphical programming language and textual programming language in the USA. Another study by Gupta (2004) focused on which programming language is good for teaching novice programmers the first programming language. Furthermore, the choice of the first programming language to teach beginners must be considered (Duke et al., 2000).

There is limited empirical evidence on the role of the C programming language as an introductory language within higher education in Tanzania. While global trends increasingly favor alternative introductory languages like python (Guo, 2014), the pedagogical value of C in this context remains underexplored. This study addresses this gap by examining student and instructor perceptions of C as a first programming language and its perceived usefulness in supporting subsequent programming learning.

1.1 Research Objectives

- i. To assess students’ perceived difficulty of learning C programming as a first programming language
- ii. To examine instructors’ perceived difficulty of teaching C programming as a first programming language
- iii. To evaluate students’ perceived transferability of knowledge gained from C to other programming languages
- iv. To address these objectives, a conceptual model was developed linking perceived ease of learning and teaching C to perceived transferability of programming knowledge. Based on this model, the following hypotheses were formulated:

1.3 Research Hypothesis

H₀₁: Students’ perceived ease of learning C programming as a first language is positively associated with their perceived ability to transfer knowledge from C to other programming languages.

H₀₂: Instructors’ perceived ease of teaching C programming as a first language is associated with their perceptions of students’ ability to transfer knowledge from C to other programming languages.

Figure 1 represents the conceptual model for this study.

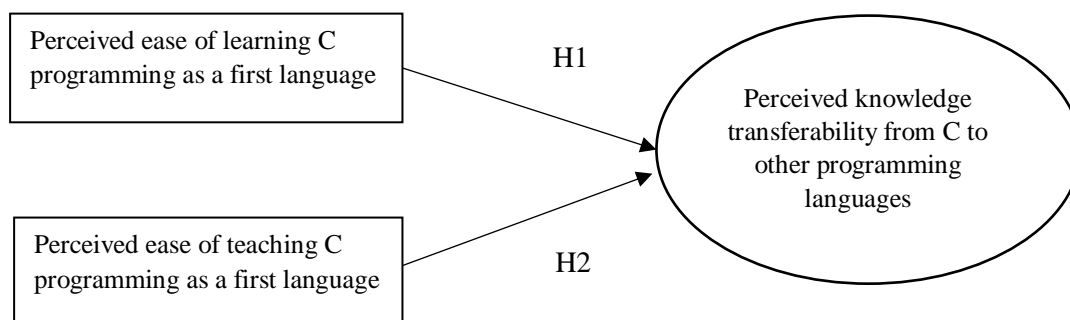


Figure 1
Conceptual Model

II. LITERATURE REVIEW

2.1 Theoretical Review

There have always been concerns among computing education researchers concerning the appropriateness of an introductory programming language used for novice programmers. One’s first programming language can be an influential factor in one’s style, techniques, coding practices, organization of source code, quality of source code, and

one's future attitude toward programming languages (Robins et al., 2003; Xinogalos et al., 2018). There are discussions, like those stated by Gupta (2004) and Thun and Eckerdal (2019), which suggest that one's first programming language affects how one thinks about programming and the development of one's computer science thinking skills.

Some criteria, based on theories, have been proposed for selecting an appropriate introductory programming language. Duke et al. (2000) suggest several principles of selecting programming languages for introductory programming courses; these principles are: simplicity, regularity, orthogonality, development support, target audience, and language coverage. According to Goosen (2004), principles for selecting a first programming language include: logically thinking, programming skills, easy programming environment for learner, affordability, general purpose and wide coverage of topic.

The characteristics of a programming language may also affect novice programmer's learning. Excessive verbosity, shortness, and complicated syntax and difficulty identifying the types of variables are considered in the programming languages making the learning experience of new learners tedious (Gupta, 2004; Mbise, 2021). In learning programming, discussions have also considered the transference of programming knowledge. According to Espinal et al. (2022), knowledge of the first programming language should enable the transfer to a professional programming language. Xu et al. (2019) point out that visual or block-based programming environments facilitate easier programming by enabling dragging-and-dropping, whereas professional text-based languages require strict use of syntax and program structure.

The C programming language is often seen as the backbone of computer science because it inspired the development of numerous modern programming languages. C++ derived from C programming (Kernighan & Ritchie, 2017). Many current programming languages, such as Java, Python, PHP and JavaScript, derived syntactical and structural properties from the C programming language (Deitel & Deitel, 2022). The language Java inherited several concepts in its syntax from C via C++ (Horstmann, 2021).

2.2 Empirical Review

There have been various studies evaluating the impact of introductory programming languages and there have been varying conclusions. A study was conducted by Chen et al. (2019) on over 10,000 US undergraduate students across 118 US higher education institutions, finding that their academic performance was significantly influenced by the introductory programming language they were introduced to with a statistically significant higher grade in students who started with a graphical programming language than text-based languages. A similar result was found by Noone and Mooney (2018), where students who learned visual programming languages obtained a higher pass rate than those taught text-based programming languages.

The ability to transfer knowledge from one programming language to another has also been addressed, with Espinal et al. (2022) noting students struggled to progress onto professional text-based programming languages when first learning with block-based languages, and Xinogalos et al. (2018) looking at C-like and Pascal-like languages with university students in Greece, Czech Republic and Serbia and finding students using C-like programming languages encountered less difficulty in subsequent programming courses than those using Pascal-like languages.

There has been a vast amount of research focusing on enhancing C programming teaching and learning through technological interventions. Fu et al. (2017) created a real-time analytics system to aid in the detection and correction of errors when students were learning C, while Daungcharone et al. (2019) observed an improvement in motivation and grades when students used a mobile game-based compiler for C programming, and Sun and Hsu (2019) determined that a smart eye-tracking feedback system improved student self-efficacy and learning performance in C programming courses.

However, there is a gap in literature regarding how students and lecturers in higher learning institutions in Tanzania perceive the use of C programming as a first programming language. Hence, this research adds to this limited field by providing context-specific data in relation to perceptions of programming knowledge transferability from C to other programming languages in Tanzania.

III. METHODOLOGY

3.1 Research Design

This study used a quantitative research approach using an online questionnaire as a method of data collection. The quantitative approach helped researchers to produce numerical data for easy interpretation of results (Kothari & Garg, 2019). Additionally, the researchers chose an online questionnaire because it offers an effective, quick, and low-cost means of data collection (Mkumbo & Mbise, 2022; Newhart & Patten, 2023).

3.2 Study Area

This research study was conducted at the Institute of Accountancy Arusha (IAA) and College of Business Education (CBE), in Tanzania using a case study research design. The researchers chose these two higher education

institutions because they use C programming language as the first language in introductory programming courses (Mbise, 2021). In addition, these institutions offer similar programmes and are regulated by the same authority, The National Council for Technical and Vocational Education and Training (NACTVET). Also, researchers opted for a case study because it provided an opportunity for a deep investigation of the role of C as a first programming language for higher education students (Creswell & Clark, 2017).

3.3 Target population

The target population of this study consisted of 158 second- and third-year Bachelor of Information Technology students enrolled at Institute of Accountancy Arusha and College of Business Education, where C programming is taught as the first programming language. In addition, the study included 50 lecturers from various higher education institutions in Tanzania who have experience teaching introductory C programming.

3.4 Sampling Procedure and Sample Size

Purposive sampling was used in selecting participants for this study. A purposive sampling method was considered appropriate in that the study targeted specific respondents who were knowledgeable and experienced in the usage of the C programming language as an introductory programming language.

For students, a comprehensive sampling approach was used. All 158 students in year two and year three in Bachelor of Information Technology at Institute of Accountancy Arusha and College of Business Education were targeted, as they have had the opportunity to have done an introductory programming course using the C programming language.

As for the lecturer respondents, snowball sampling technique was used. A couple of lecturers who were known by the researchers that teach an introductory programming course using C programming language were invited to answer the online questionnaire. Thereafter, these were asked to help the researchers identify and refer other lecturers who have similar experiences in teaching introductory programming courses from other higher education institutions in Tanzania. Snowball sampling technique was thought suitable since lecturers with specific experiences were easily reachable by the help of this technique.

3.5 Data Collection Instruments and Procedure

Data was gathered by use of two structured questionnaires; a students' questionnaire and a lecturers' questionnaire. All scales were scored using a five-point Likert-style scale from 1 (Strongly Disagree) to 5 (Strongly Agree). The questionnaire for students aimed to evaluate perceptions about the ease of learning C programming as the first programming language and its transferability value to other programming languages. The questionnaire for lecturers aimed to evaluate perceptions about the teaching ease and the perceived knowledge transferability from C programming to other programming languages.

The study employed an online questionnaire for data collection. A link to a Google Forms document containing the students' questionnaire was widely distributed to all students via class representatives on the WhatsApp application, thus facilitating ease of participation. The link to the lecturers' questionnaire was circulated to lecturers via direct distribution, followed by a follow-up telephone call to enhance participation. The students and lecturers' questionnaires yielded a total of 96 and 42 responses respectively.

3.6 Data Analysis

The data collected was analyzed using SPSS Statistics Software. Descriptive statistics were computed to report the respondents' perceived view on issues related to this research. Further, correlation and regression analysis were conducted to explore the relationship between perceived ease of learning/teaching C programming and perceived transferability to other programming languages, not to establish the cause and effect relationship between variables.

3.7 Ethical Consideration

Ethical conduct and protection of rights of respondents throughout the course of study. Prior to the data collection the respondents were clearly informed regarding the intention and objectives of the study. Participation to the study was voluntary and the respondents had the right to participate or withdraw from the study at any time and not penalized for the action.

Prior to filling the questionnaire informed consent was obtained from respondents. The researchers made a promise to the respondents that the collected information would only be used for academic purposes. Anonymity and confidentiality was ensured and participants did not mention their name nor have their personal identities taken in the questionnaires and were not included in the final study report.

In addition, the online questionnaire was designed to avoid any harm and discomfort to respondents. The collected data was protected and accessed only by the researchers conducting the study.

3.8 Validity and Reliability

Exploratory factor analysis was conducted to assess the validity and reliability of the scales employed in this study, along with the reliability testing of individual scale items. For the easy to teach C programming scale, one dominant factor was extracted via principal components extraction in the factor analysis, with all items being loaded on the same factor whereby items with weak loadings were dropped (retained factor loadings from 0.84 to 0.95). A Cronbach's alpha value of 0.71 for the internal consistency of the easy to teach C programming scale indicated that the construct of ease to teach C programming had satisfactory internal consistency. Similar to the easy to teach C programming scale, the easy to learn C programming scale was also found to be a unidimensional measure (factor loadings ranged from 0.86 to 0.96, Cronbach's alpha value of 0.90), which shows high construct reliability. For the scale of perceived programming knowledge transferability, all items loaded on the same factor suggesting that the items intended to measure a single underlying construct with factor loadings of the retained items ranging from 0.66 to 0.85. One low-loading item was excluded for the improvement of construct validity, leading to a Cronbach's alpha of 0.69, suggesting that this scale had acceptable reliability. Overall, the measurement scales demonstrate acceptable levels of validity and reliability; however, moderate reliability in some constructs and potential noted theoretical overlap between constructs should be considered when interpreting the findings.

IV. FINDINGS & DISCUSSION

4.1 Findings

4.1.1 Descriptive Results of Students' and Lecturers' Perceptions

The study first examined students' and lecturers' perceptions of C programming as a first programming language, focusing on ease of learning, ease of teaching, and perceived transferability of knowledge to the learning of other programming languages.

Table 1

Students' Perceptions of Ease of Learning C Programming

Variable	N	Mean	STD
Learning C programming as a first language was easy	96	3.71	1.34
Writing programs in C programming as a first language was easy	96	3.77	1.29
It was easy to understand programming using C as a first programming language	96	3.69	1.15

The results in Table 1 indicate that students generally reported moderate agreement that learning C programming as a first language is easy. The mean scores ranged from 3.69 to 3.77 (SD = 1.15–1.34), indicating generally positive but somewhat varied perceptions among respondents. The highest agreement was observed for the ability to write programs in C (M = 3.77, SD = 1.29), while the lowest was related to understanding programming using C (M = 3.69, SD = 1.15).

Table 2

Lecturers' Perceptions of Ease of Teaching C Programming

Variable	N	Mean	STD
I find it easy to teach C programming	42	3.83	1.34
I find it easy to teach C programming to students as a first language	42	3.55	1.48
Teaching C to beginners is easy	42	3.48	1.49

Table 2 presents lecturers' perceptions regarding the ease of teaching C programming as a first language.

Table 3

Perceived Transferability of Knowledge from C Programming

Variable	Students (N=96)		Lecturers (N=42)	
	Mean	STD	Mean	STD
Knowledge gained from C programming is useful when learning other programming languages	3.63	1.27	3.52	1.47
Concepts from C may help on understanding of other languages	3.61	1.15	3.55	1.23
Other programming languages can easily be learnt after studying C programming as a first language	3.65	1.32	3.34	1.19



The results indicate moderate agreement overall, with mean values ranging from 3.48 to 3.83 (SD = 1.34–1.49). The highest agreement was observed for the general statement that teaching C programming is easy (M = 3.83, SD = 1.34), while lower agreement was observed for items related to teaching effectiveness with beginners (M = 3.48, SD = 1.49), indicating variability in teaching experience perceptions.

Table 3 depicts that the mean response of students and lecturers to their agreement level that knowledge from learning C language is transferable to other programming languages falls in moderately agree response which mean scores of students range from 3.61-3.65 (SD = 1.15-1.32) and range of lecturers are 3.34-3.55 (SD = 1.19-1.47) respectively. Student gave a little higher responses than lecturers where they agreed more to the ease learning of other programming languages after learning C language. In a nutshell, based on the findings in Table 1, the respondents (students) believe that learning programming using C language as first language of programming is relatively easy to learn and can provide a good foundation in learning other programming languages while in the case of lecturer also believe that C programming is relatively easy to teach among student with highest variability. For the transferability, both students and lecturers moderately agree that knowledge gained through C programming can be transferred to other programming languages.

4.2 Regression Analysis

Regression analysis was performed on the relationship between perceived ease of learning and teaching C programming, and the perceived transferability of programming knowledge from C to other languages. Two regression models were estimated. Model 1: ease of teaching on transferability based on data from lecturers, and Model 2: ease of learning on transferability based on data from students. Two models were used due to different respondent groups giving differing accounts (teaching and learning), and therefore data could not be put into a single regression without model misspecification.

Model 1 Effect of Ease of Teaching on Transferability

Table 4

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
0.88	0.77	0.77	0.43

The model summary on table 4 indicates that 77% of the variance in perceived transferability is explained by ease of teaching C programming ($R^2 = 0.77$), suggesting a strong association between the variables.

Table 5

ANOVA Results

	Sum of Squares	df	Mean Square	F	Sig.
Regression	24.36	1	24.36	134.69	0.000
Residual	7.24	40	0.18		
Total	31.60	41			

The ANOVA results show that the model is statistically significant, $F(1, 40) = 134.69$, $p < 0.001$, indicating that ease of teaching significantly predicts perceived transferability.

Table 6

Regression Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	1.50	0.19	0.00	8.07	0.000
Easy of Teaching	0.58	0.05	0.88	11.61	0.000

The results indicate that ease of teaching C programming is a strong and statistically significant positive predictor of perceived transferability ($\beta = 0.88$, $p < 0.001$).

Model 2: Effect of Ease of Learning on Transferability

The model explains 37% of the variance in perceived transferability ($R^2 = 0.37$), indicating a moderate relationship. The findings were presented in Table 7.

Table 7*Model Summary*

R	R Square	Adjusted R Square	Std. Error of the Estimate
0.61	0.37	0.36	0.57

Table 8

ANOVA Results

	Sum of Squares	df	Mean Square	F	Sig.
Regression	17.65	1	17.36	54.51	0.000
Residual	30.44	94	0.32		
Total	48.08	95			

The ANOVA results indicate that the model is statistically significant, $F(1, 94) = 54.51, p < 0.001$.

Table 9*Regression Coefficients*

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	2.28	0.19	0.00	11.79	0.000
Easy of Learning	0.37	0.05	0.61	7.38	0.000

The results show that ease of learning C programming is a statistically significant positive predictor of perceived transferability ($\beta = 0.61, p < 0.001$). Generally, results of the regressions indicate that both ease of learning programming, and ease of teaching programming are significant predictors of transferability. This relationship however varies between the two sets of results. The model utilizing lecturer data (ease of teaching) appears to account for significantly more variance than the model utilizing student data (ease of learning, $R=0.37$), with $R=0.77$ for lecturer data. Student and lecturers reported similarly positive attitudes towards C programming as a first language in their responses (mean values of 3.36 to 3.83, respectively, with large standard deviations).

4.3 Discussion

The outcomes of this research, when taken overall, are generally in line with the established body of work that argues that the selection of the first programming language used in computer science education is an important decision (Robins et al., 2003; Xinogalos et al., 2018; Thuné & Eckerdal, 2019). Students in this study expressed moderately positive perceptions, which aligns with the suggestion that a programming language used early in learning helps shape students' confidence and perceived ability to transfer their learning. More specifically, the significant positive relationship between perceived ease of learning C and perceived transferability aligns with findings by Xinogalos et al. (2018), who found students to report fewer learning difficulties with C-like programming languages later in their computing courses. Similarly, Chen et al. (2019) also reported that the introductory programming language could affect student performance, and that it is therefore an important choice. The logic here is likely that if students feel that the programming language is easier to learn, they feel more confident and believe they can apply what they learn to another programming language.

It is worth mentioning that, while there was a significant positive relationship between perceived ease of teaching from the lecturer dataset, these are the lecturers' perceived ease of teaching rather than the students' learning experience. This is different to previous work, which often emphasized the role of pedagogical design and instructional methods in learning transfer (Parker et al., 2006; Espinal et al., 2022). This study did not measure results of the instructional design, teaching methodology and student engagement. Because of this, lecturer perceptions of ease of teaching may not fully encapsulate the effect on learning.

Another important finding of this research is that student and lecturer responses vary largely; which is reflected by the moderately large standard deviations. This is a clear indication that there is no universally held perception of C among students (and lecturers) and supports the idea that it can be difficult for students to learn an unfamiliar language due to syntax complexity and abstraction levels (Gupta, 2004). While C is a foundational language and an inspiration for many other contemporary programming languages (Kernighan & Ritchie, 2017; Deitel & Deitel, 2022), this research seems to provide some evidence against a perceived universal ease of learning.

It is also important to remember that these are students' and lecturers' perceived abilities and not measured learning outcomes. Given that many of these relationships were strongly significant, it is highly likely that measurement

context played a significant role in the data. Specifically, respondents likely answered more favorably (or negatively) depending on whether they had just answered a question regarding another measure. This cautions us against establishing causal implications based on these results. Overall, the results of this study lend support to the existing body of work emphasizing the importance of the first programming language while still showing that programming learning is influenced by several intertwined factors. Consistent with Thuné and Eckerdal (2019), it is clear that student learning and their attitudes toward programming are a complex construct involving a multitude of factors.

V. CONCLUSION & RECOMMENDATIONS

5.1 Conclusion

The objective of this study was to explore how C programming when learned as a first language is perceived by students and lecturers at the higher education level in Tanzania in relation to how it helps students to learn other programming languages. It appears that both students and lecturers are neither so positive nor so negative about C as a first language. Also, it appears there is diversity on this perception as the variation in the answers provided by the students and lecturers was high. The students' perceived ease of learning C programming was found to be highly correlated with students' perceived ease of learning other programming languages whereas the lecturers' perceived ease of teaching C programming was not found to be significantly correlated with the perceived ease of learning other programming languages. This finding suggests that the perceived transferability was mainly attributed to the students' own learning experiences. It is important to note that this study found associations rather than cause and effect relationships in term of perceptions. In essence, the study contributes to computing education research in terms of producing empirical evidence from higher education in Tanzania which is one of the countries, that is under-represented when it comes to introductory programming language use. Also, the study moves from discussing causal "*impact*" of a first programming language, to "*perceived transferability of programming knowledge*" in terms of how to study this using data collected from the participants, and is able to provide a much more valid approach than earlier studies that have focused on claimed impact in a learning process. It also suggests that the students' perception of the ease of learning C programming has a stronger effect on their perception of being able to understand other programming languages compared to the lecturer's perception of teaching ease. Finally, taking both students and lecturers' views provides a more complete picture of introductory programming practices, and provides a starting point for other similar studies which can incorporate measures of performance and comparative designs.

5.2 Recommendations

Drawing from the conclusions of this study, the following recommendations are offered to higher learning institutions, curriculum developers, lecturers and future researchers in computing education.

First, institutions of higher learning in Tanzania should continue strengthening introductory programming instruction by implementing student-centered learning methods that promote effective students' learning experience during introductory programming instruction. As perceived ease of learning C programming was positively related with the perceived transferability of programming knowledge, emphasis should be given to improving the learning environment, practical programming activity, coding and problem-solving exercise.

Second, lecturers should engage interactive and facilitating learning activities such as guided programming exercise, peer learning activity, debugging sessions, project based learning and formative assessment. These learning activities may reduce students' programming anxiety and enhance their confidence in transferring programming knowledge to another programming language.

Third, policy makers and curriculum designers in higher learning institutions should critically review the selection of introductory programming language, considering not only relevance to the industry but also learners' experience, understanding of concepts and transferability of programming skills. While C programming is shown in this study as having a positive relationship with transferability of knowledge, curriculum designers must ensure that introductory programming course design meet the present need of both technology and learning environments.

Lastly, future research work in computing education could expand this research by investigating objective programming performance measurements, applying a longitudinal study design and comparative study to other introductory programming languages such as Python, Java and C++. Such studies can provide a more in-depth understanding of how introductory programming languages affect students' learning and future development of programming expertise.

Declaration of Interest

The authors declare that they do not have any known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding Declaration

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

REFERENCES

- Chen, C., Haduong, P., Brennan, K., Sonnert, G., & Sadler, P. (2019). The effects of first programming language on college students' computing attitude and achievement: A comparison of graphical and textual languages. *Computer Science Education*, 29(1), 23–48. <https://doi.org/10.1080/08993408.2018.1547564>
- Creswell, J. W., & Plano Clark, V. L. (2017). *Designing and conducting mixed methods research* (3rd ed.). SAGE Publications.
- Daungcharone, K., Panjaburee, P., & Thongkoo, K. (2019). A mobile game-based C programming language learning: Results of university students' achievement and motivations. *International Journal of Mobile Learning and Organisation*, 13(2), 171–192. <https://doi.org/10.1504/IJMLO.2019.098184>
- Deitel, P., & Deitel, H. (2022). *C how to program* (9th ed.). Pearson.
- Dmitrović, S. (2021). *Modern C for absolute beginners: A friendly introduction to the C programming language*. Apress.
- Duke, R., Salzman, E., Burmeister, J., Poon, J., & Murray, L. (2000). Teaching programming to beginners—Choosing the language is just the first step. In *Proceedings of the Australasian Conference on Computing Education* (pp. 79–86). <https://doi.org/10.1145/359369.359381>
- Edefe, M. (2020). *Computer applications: The beginner's guide*. Amazon Digital Services LLC.
- Espinal, A., Vieira, C., & Guerrero-Bequis, V. (2022). Student ability and difficulties with transfer from a block-based programming language into other programming languages: A case study in Colombia. *Computer Science Education*, 33(4), 1–33. <https://doi.org/10.1080/08993408.2022.2079867>
- Fu, X., Shimada, A., Ogata, H., Taniguchi, Y., & Suehiro, D. (2017). Real-time learning analytics for C programming language courses. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference* (pp. 280–288). <https://doi.org/10.1145/3027385.3027407>
- Gonzalez, A. J. (2021). *Computer programming in C for beginners*. Springer International Publishing.
- Goosen, L. (2004). *Criteria and guidelines for the selection and implementation of a first programming language in high schools* (Doctoral dissertation, North-West University). <https://repository.nwu.ac.za/handle/10394/226>
- Guo, P. J. (2014, July 7). Python is now the most popular introductory teaching language at top U.S. universities. *Communications of the ACM Blog*. <https://cacm.acm.org/blogcacm/python-is-now-the-most-popular-introductory-teaching-language-at-top-u-s-universities/>
- Gupta, D. (2004). What is a good first programming language? *XRDS: Crossroads, The ACM Magazine for Students*, 10(4), 1–15. <https://doi.org/10.1145/1027313.1027320>
- Horstmann, C. S. (2021). *Core Java: Fundamentals* (Vol. 1, 12th ed.). Pearson.
- Kernighan, B. W., & Ritchie, D. M. (2017). *The C programming language*. CreateSpace Independent Publishing Platform.
- Kochan, S. G. (2014). *Programming in C* (4th ed.). Addison-Wesley Professional.
- Kothari, C. R., & Garg, G. (2019). *Research methodology: Methods and techniques* (4th ed.). New Age International.
- Mbise, K. S. (2021). The role of IT professional certifications in IT instructors' teaching quality. *International Journal of Education and Development Using Information and Communication Technology*, 17(1), 176–187.
- Mkumbo, D. R., & Mbise, K. S. (2022). The influence of social media advertisements on fast-food consumption. *The Journal of Informatics*, 2(1), 22–32.
- Newhart, M., & Patten, M. L. (2023). *Understanding research methods: An overview of the essentials*. Taylor & Francis.
- Noone, M., & Mooney, A. (2018). Visual and textual programming languages: A systematic review of the literature. *Journal of Computers in Education*, 5(2), 149–174. <https://doi.org/10.1007/s40692-018-0101-5>
- Parker, K. R., Ottaway, T. A., & Chao, J. T. (2006). Criteria for the selection of a programming language for introductory courses. *International Journal of Knowledge and Learning*, 2(1–2), 119–139. <https://doi.org/10.1504/IJKL.2006.009683>
- Perry, G., & Miller, D. (2017). *C programming absolute beginner's guide* (3rd ed.). Que Publishing.
- Prokop, Y., Trofimenko, E., Severin, N., & Bukata, L. (2019). An analysis of criteria for choosing a first programming language in universities. In *Proceedings of the 15th International Conference on ICT in Education, Research and Industrial Applications* (pp. 420–425).
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172. <https://doi.org/10.1076/csed.13.2.137.14200>
- Seacord, R. C. (2020). *Effective C: An introduction to professional C programming*. No Starch Press.



- Sun, J. C.-Y., & Hsu, K. Y.-C. (2019). A smart eye-tracking feedback scaffolding approach to improving students' learning self-efficacy and performance in a C programming course. *Computers in Human Behavior*, 95, 66–72. <https://doi.org/10.1016/j.chb.2019.01.036>
- Szeliski, R. (2022). *Computer vision: Algorithms and applications*. Springer Nature.
- Szuhay, J. (2022). *Learn C programming: A beginner's guide to learning the most powerful and general-purpose programming language with ease* (2nd ed.). Packt Publishing.
- Thuné, M., & Eckerdal, A. (2019). Analysis of students' learning of computer programming in a computer laboratory context. *European Journal of Engineering Education*, 44(5), 769–786. <https://doi.org/10.1080/03043797.2018.1544609>
- Xinogalos, S., Pitner, T., Ivanović, M., & Savić, M. (2018). Students' perspective on the first programming language: C-like or Pascal-like languages? *Education and Information Technologies*, 23(1), 287–302. <https://doi.org/10.1007/s10639-017-9601-6>
- Xu, Z., Ritzhaupt, A. D., Tian, F., & Umapathy, K. (2019). Block-based versus text-based programming environments on novice student learning outcomes: A meta-analysis study. *Computer Science Education*, 29(2–3), 177–204. <https://doi.org/10.1080/08993408.2019.1565233>