Students' Perception of the Effect of Cognitive Factors in Determining Success in Computer Programming: A Case Study

Jotham Msane[1], Bethel Mutanga Murimo[2], Tarirai Chani[3]
Department of Information and Communication Technology
Mangosuthu University of Technology, MUT
Umlazi, South Africa

***Abstract*—The reliance on science and technology by both countries and corporate entities is increasingly evident as the evolving trend of digitization not only pervades every facet of life but also assumes a dominant role. Correspondingly, the significance of producing competent computer science and information technology (IT) graduates becomes highly imperative. Already, in most developed and developing countries, there has been an increasing demand for these competencies such as network engineers, programmers, and other IT-related specialists. Although these competencies are equally valuable, programming skills constitute the core of the strength of every other IT-related competence. Nevertheless, programming is reported in the literature to be one of the most difficult courses to students. Moreover, the level of performance in programming is said to be significantly low with an attendant high rate of students' dropout. There is a concerted research effort toward addressing the challenge of poor academic performance by attempting to answer the question of what factors affect academic performance in general. However, there is scanty literature on the factors that affect the ability to understand the concept of programming in specific. This paper, therefore, reports a case study investigation of students' perception of the effect of cognitive factors as the determinant of success in computer programming. The findings showed that performance in introductory programming is impacted by a range of interrelated cognitive factors including self-efficacy and the love for technology.**

Keywords—Cognitive factors; performance; programming; self-efficacy

I. Introduction

Under-performance and a high dropout rate of undergraduate students remains a major problem that characterized higher education in South Africa [1], [2]. This predicament has tremendous adverse implications on both the private and public sectors of the economy because students are the potential assets of the country's economy as they transfer skills from the Universities to the industry. The government commits huge budgetary allocations annually through the Department of Higher Education and Training as part of a broad national strategy to improve the quality of life of South Africans by supporting the post-school education and training system.

Among others, science and technology disciplines record the lowest performance and highest dropout rates. Within South African Universities of Technologies as well as other countries, this ugly trend of poor performance is very pronounced among first-year students, especially in introductory programming courses [3]. Moreover, under-performance in programming has far-reaching implications. For instance, it can undermine the capacity to meet the global trend of a shift in skill demand in response to the advent of the Fourth Industrial Revolution (4IR) [4], [5]. This revolution hinges on digital technology as the pivot and driver of global innovation because it ushers in a paradigm shift from the hardware-centered to software-centered technology [5].

Whether for the student, researcher, or industrialist, the transition to 4IR, promises unlimited prospects in the emergent fields of Internet of Things, cloud computing, artificial intelligence, machine learning, and big data analytics. From a labour market perspective, these prospects present an ever-growing global demand for the IT skills necessary to steer the resultant digital economic, social, and other innovative systems in both developed and developing countries [6], [7]. This realization essentially emphasizes the indispensability of computer programming skills in an increasingly IT-dominated world. Therefore, it becomes highly imperative for universities that offer IT courses to strengthen their capability to produce graduates with sound programming skills.

Consequently, the need to address underperformance in programming has gained wide research attention with much emphasis on identifying the predictor factors [8], [9], [10].

However, with the global average success rates in introductory programming courses estimated to be 67 percent in [11], the need to investigate why several students find introductory programming difficult [3] to understand remains highly imperative.

In this study, therefore, the perception of students on the effect of cognitive factors as a predictor of academic performance in computer programming was investigated. The study contributes toward enhancing curriculum development for South African Universities of Technologies and providing useful insights for crafting intervention programs to assist at-risk students. The study used a case study of second-year Information Technology students at a South African University of Technology.

The rest of the paper is structured as follows: the review of relevant works in presented in Section II and the cognitive learning theory briefly explored briefly explored Section III. The study's methodology and case study description are offered Sections IV while the findings are reported in Section V. The paper in concluded in Section VI.

II. Literature Review

With the emergence of Big data analytics and machine learning tools, several models are being designed to predict academic performance in diverse fields as reported in [12]–[14]. Such models are grounded on the rationale that academic performance is the resultant effect of certain determinable variables or factors. As researchers seek to understand the phenomenon of academic performance, these factors have been the key focus of various investigations. Therefore, this review is strucrured based on the categories of factors as reported in the literature, namely, academic factors, cognitive factors, and psychosocial factors.

A. Academic Factors

Factors such as previous academic achievement, mathematical abilities, prior experience, and study skills have widely alluded by many studies to have a strong correlation with academic performance. In their work to rate predictor factors, authors in [15] maintained that previous academic achievement is a single best predictor of student success at the university level. According to the authors, a student who obtained good results in high school is likely to perform better in the university. This finding is also supported in [16]. In a similar effort reported in [17], students' previous achievement was found to best predict their academic performance when combined with self-efficacy.

Another work in [18] narrowed the focus to investigate the joint effect of prior programming background and self-efficacy on undergraduate students' success in programming. Based on their findings, they concluded that prior programming does not affect performance directly, but it increases one's self-efficacy for programming hence increase performance. In the same vein, the relationship between mathematics and programming was investigated in [19]. The study confirmed that students' mathematics ability was strongly related to their programming performance. Similarly, other studies such as using a mixed-method approach, the work in [20], [21] all corroborate the strong correlation between mathematics abilities and performance in programming. Authors in [21] specifically stated, "The data shows that if the mathematics subject result is good, then the programming result is also good".

B. Psychosocial Factors

There have been consistent reports in the literature that psychosocial factors impact academic success. Psychosocial factors such as students' social integration, career orientation, commitment to the study, social support, psychological health, amongst others have been investigated. According to [22] Tinto's theory indicate that a student tends to persist and perform better if integrated well to the university. This claim is also strongly supported by the findings reported in [23]. But contrary to Tinto's model, the work in [24] revealed that the commitment of students to the university's social activities results in poor performance. This may be true for courses like programming which require a lot of time. Too much involvement of students in the university social groups may occupy the very time required for practice since computer programming requires practice in and out of the classroom. However, the authors further concluded that students who like the course they study are often associated with clear objectives and are likely to obtain good results.

C. Cognitive Factors

There is extensive research on potential effect cognitive factors on academic success. The majority of these studies focus more on first-year students. The reason for this is because according to [25], the students tend to drop out of the university during the first academic year or before the second year of study as a result of poor academic achievement. Poor performance or dropouts of students is a concern to the academic institutions. Numerous studies have identified several cognitive factors to influence academic achievement. These factors include but not limited to mental models, self-efficacy beliefs, motivation, and personality traits. However, self-efficacy and motivation have been for a long time, indicated as having a strong impact on the academic success of students.

Studies on self-efficacy have constantly asserted that students with high self-efficacy are likely to achieve good results at university than those with low self-efficacy. The work in [26] noted that self- efficacy has been proven across the literature as the strongest predictor of academic performance. The author of the work in [27] is also convinced that self- efficacy is a powerful predictor of academic performance. According to the author, every action one could take starts in the mind. Thus, "people's beliefs in their efficacy influence the type of anticipatory scenarios they construct and rehearse".

The work in [28] studied the effect of self-efficacy, mathematics, self-concepts, perceived usefulness of mathematics, and prior experience with mathematics on the prediction of problem-solving. Their study revealed that mathematics self-efficacy was the best predictor of problem-solving. The work did not examine the joint effect of these factors because the findings in [29] support the fact that self-efficacy tends to increase for one with prior experience. A study conducted in [30] examined the influence of combining self-efficacy, mental model, and prior introductory programming experience and it was discovered that self-efficacy of a student with previous programming experience tends to increase significantly throughout the programming course in the first-year level.

III. Cognitivism and Cognitive Factors

Cognitivism as a learning theory explains how mental processes are influenced by both external and internal factors to produce learning in an individual. The theory is centred on the idea that learning occurs when individuals process information they receive, rather than merely responding to stimuli as illustrated in Fig. 1. That is, the individual takes in the stimulus, processes it in their mind, and then acts upon the stimulus, which alludes to the fact that in the cognitive learning process, new knowledge is built upon prior knowledge.

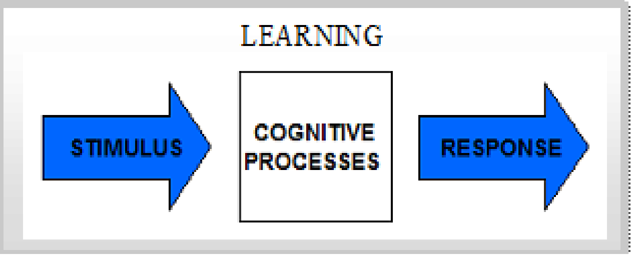

Fig. 1. The Cognitive Learning Process.

The mental processes that leads to learning consist of several elements of the individual which include attention, observation, perception, reasoning, organizing, memory, and forming generalizations. These elements or factors represent those characteristics of a person that affect the way they learn and perform [31].

Essentially, cognitive factors are intrinsic characteristics, therefore, they cannot be measured qualitatively. However, as learned in the literature, cognitive ability is associated with some perceived behavioral attitudes also referred to as cognitive predictors. Based on this understanding, the data for this study was obtained by measuring students' perceptions of the cognitive predictors that influence performance in introductory programming.

IV. Case Study Description and Methodology

This case study was structured in a manner that allows the researches to explore the effect of cognitive factors on a total of 20 selected second-year students. The study population was then categorized into two subgroups based on the students' first and second semesters average performances in their first year. Each subgroup was further partitioned into two equal focus groups. To ensure confidentiality and avoid stigmatization, the basis of the categorization was not disclosed and neither of the two groups knew about the existence of the other. More importantly, it was presumed that participants may not be willing to give out honest information should they know that the investigation equally involves some other students who might know them.

The first subgroup consists of ten students who had an average score of less than 50% in their first and second semester year one introductory programming. The data was officially obtained from the department of information and communication technology by one of the researchers who facilitates the course. Scores within that range depict a poor to fair performance. This categorization condition was intended to enable the us to evaluate the students' perception on how cognitive factors may relate to their performance. On the other hand, the second subcase comprised of another ten students whose average score is 50% or higher. Similarly, this score range of identified students of good to excellent performance.

Because this study investigates a real-world problem, the methodology is designed to ensure flexibility in the process of gaining concrete and in-depth contextual knowledge through two data collection tools – unstructured questionnaires and focus group interviews. Qualitative data may be broader and richer [32], but it may also suffer the deficit of being less precise. Consequently, the strategies of data triangulation and prolonged involvement strategies to enhance precision and strengthen the validity of the study as outlined by Robson and other authors [33] were utilized in the study. Data triangulation was achieved by using more than one tool to collect the same data on different occasions, which gives the researchers multiple perspectives towards the studied population thereby providing a broader picture. Also, the study leveraged the benefit of prolonged involvement. The long-term relationship that already exists between the participants and researchers, who are both lecturers in Information and Communication Technology Department, enabled the investigators to understand how participants interpret terms used in the study and created an atmosphere of trust that ensured participants spent more time providing data.

Ultimately, the first phase of data collection involved the use of unstructured questionnaires to allow interviewees to articulate their thought unrestricted. Then followed by unstructured questionnaire with open-ended. The funnel interview model was adopted. With this model, the objectives of the interview were first presemted and the manner the data from the interview would be used was explained. Next, participants were asked open questions that then led to specific questions around cognitive and cognitive performance factors.

To ensure quality and engaging interaction and ease note-taking, the study population was partitioned into four focus groups and interviewed each group separately for 15-30 minutes. By interviewing each interviewee more than once, the researchers aim at gathering data that is both detailed and rich in context. The patterns that emerged from these interviews comprise the bulk of the data collected, transcribed, coded, and then qualitatively analyzed.

V. Discussion of Findings

From the data obtained, this study identified five cognitive predictors that influence students' performance in introductory programming. This discussion of this study's findings are, therefore, presented based these factors.

A. *Academic Association*

*T*his study revealed that most students believe that one's ability to associate academically with other students has a positive impact on academic achievements. The respondents strongly believe in the power of peers assisting each other. One respondent even went as far as starting, "I would not have pass intro. Programming but not for the support I received from my course mate who was very good in programming was my greatest nightmare – I felt like dropping out".

I can, therefore, be concluded that by helping one another in the classroom and outside of it, students are more likely to fully understand and remember the topics done in class, especially, in the concept of programming which often knew to most students. This is strongly supported by literature which states that students learn better when they interact as peers. The study conducted in [34] studied the impact of peers in mastering mathematics. The study concluded that peer learning facilitates the process of learning more effectively than a normal classroom. The effectiveness of peer learning was further lamented by the study conducted in [35]. Although the study came to the same conclusion, but also put a lot of emphasis on the effect of the environment rather than the association.

B. Cognitive Factors

Current literature widely alludes to the fact that self-belief positively impacts students' ability to execute tasks both in the classroom and outside. This finding also applies in the context of programming as this study equally supported this fact. Of the 20 respondents, 90% of them believed so much in the power of self-efficacy.

A study conducted in [28] found self-efficacy to be a powerful driver towards the ability to perform introductory programming. Studies in other fields equally corroborate this finding. For example, the work in [36], on people working on musical projects. The study concluded that employees with better self-efficacy performed well in musical projects than employees with low self-efficacy.

However, some respondents were of the view that self-efficacy alone is not enough to master programming. Some respondents thought that hard work should accompany self-efficacy for success to be guaranteed. On the other hand, some respondents viewed hard work as a product of self-efficacy. The belief was that once someone has a high self -efficacy, automatically they tend to be hard workers hence resulting in better performance. Apart from the fact that all respondents that performed extremely well in introductory programming were found to demonstrate the attitude of self-believe, some respondents noted that their performance only started to improve when they started believing that they too can learn programming.

Consequently, it can be infered that programming requires a lot of dedicated effort in the form of regular practice and self-efficacy is key to exercising such dedication. This also submission agrees with the study in [37], which concludes that self-efficacy is strongly correlated with hard work. According to the authors, there is a strong relationship between the level of self-efficacy and the effort that students.

C. Self-Drive to Work

The respondents frequently indicated that in the course such as programming, students need to put more effort into his or her work to get good results. They further alluded that students who usually perform better are those who spend most of their time working on solving programming problems in the labs or on their personal computers. Some respondents pointed to the laziness as an enemy of success in the computer programming course. This denotes that students who are given exercises in class and make little or no attempts are likely to perform poorly in the assessment that follows thereon while those who work hard are likely to do better. The study conducted in [26] examining the potential factors affecting student success is also in support of the view that the more effort a student puts on his/her work, the higher the chances of success in a programming course.

Hardworking was a commonly mentioned term by most of the respondents. They believe that a programming course by nature is practical. Thus, success requires one to be dedicated and learn by doing. The study revealed three concepts that have been used interchangeably by respondents, effort, hardworking, time spent on programming activities as having a significant influence on academic performance. These findings are in agreement with those of [27] who also found effort and hardworking to have a significant impact on academic performance. This is even though their study was examining these factors on the success in Mathematics. However, the majority of studies have shown a strong relationship between mathematics and computer programming due to their practical and problem-solving nature. It is therefore it bodes well to consider their argument about the findings of this study.

To improve student performance, the instructors must ensure that all students in the programming class are encouraged and monitored. Students should be monitored so that they do all their given class exercises or tutorials before given corrections as that will enforce everyone to practically learn.

D. Motivation

Motivation has been indicated in the literature to have a strong influence on academic performance and performance in the workplace. This study also revealed similar results. Motivation has been singled-out by almost all respondents as the key factor influencing one's academic performance. However, some respondents noticed that motivation does not directly influence academic performance. This is in line with the study conducted in [27]. The respondents enlightened that even the weak student if he or she is motivated is likely to succeed in programming. This is because the motivated individual has the drive to work hard and if one is motivated tend to never give up instead work even harder when faced with difficulties in the programming task. Even good students tend to perform poorly as the motivation diminishes.

The findings in [28] concluded that a high sense of self-efficacy tends motivating students to persevere. Furthermore, the authors also concluded that the increased perseverance results in the student putting more effort into programming.

The respondents also pointed out repeatedly that the reason for most of the students to drop out of the introductory programming course is when students fail the first few assessments resulting in them being demotivated. Theories of motivation support the view that performance decrease as the motivation decreases and increases as motivation increases.

There are two types of motivation, intrinsic motivation, and extrinsic motivation. This study further enlightened that extrinsic motivation is variable, tends to change, in the case of programming this type of motivation may change as difficulties and challenges arise during the course. The respondents also pointed out that self-motivation (intrinsic-motivation) is better than extrinsic motivation. While, self-motivated individuals who are also known as mastery-oriented individuals, they also tend to put more effort when the results are not in their favor.

E. The Love for Technology

Love of technology, and specifically, computer programming was frequently mentioned by most respondents, as influencing academic performance in computer programming. Studies that examined the relationship between computer game playing and success in programming advocate that students who like to play computer games are likely to succeed in programming. In this study found that students who love programming tend to be self-motivated and seek to understand than just doing it for the sake of obtaining a qualification. This type of students tends to be hard workers and always seek to find more information than what has been covered in the classroom. Such students spend time on YouTube video tutorials and other online materials hence find useful information on their own rather than in the classroom. Some respondents further stated that IT was not their preferred career path but decided to enroll for it because of desperation. These students were found to have very low self-efficacy of programming and put very little effort into their work. The work in [29] and [30] strongly argues that students who are doing courses that are not of their preferences tend to perform poorly.

VI. Conclusion

Educationists and educational psychologists have, for a long been investigating the factors that affect the performance of students. This paper explored the perception of students on the effect of cognitive factors on academic performance. The study aimed at addressing the challenge of underperformance in introductory programming course at a South African University of Technology. The findings of the the study showed that students strongly feel that cognitive factors have a significant impact on their performance. Whereas most studies seem to lay more emphasis on students' academic history, this study suggest that academic performance is influenced by a wider range of other interconnected factors. These findings have the potential to inform the design of a broad-based curriculum that incorporate both social and psychological components. Through such inclusions, performance inhibitors that stem from psychosocial factors can be addressed.

However, this study has limitations which are acknowledged as follows: geographically limiting the study to one institution as well as focusing only on ICT students may have effect in generalizing the results. Therefore, an investigation of more varied populations of universities and programs need to be conducted to improve the quality and validity of the results. That due to the delineation of this study, there was no comparative analysis with other similar methods. However, such an analysis has the potential to trengthen the possibility of generalization. Therefore, the future study from this work will focus on using existing frameworks to compare the methods adopted in this study to others in the current literature.

References

[1] M. van Wyk, J. Christian, M. Seromo, S. du Preez, A. Legodi, and L. Erasmus, "The effective use of blended learning tools that promote success among low performing South African undergraduate students," Progressio, vol. 39, no. 1, pp. 67–88, Mar. 2018.

[2] C. W. Callaghan and E. Papageorgiou, "Personality, gender and student performance at a South African University," Africa Educ. Rev., vol. 17, no. 1, pp. 66–82, Jan. 2020.

[3] S. I. Malik and J. Coldwell-Neilson, "A model for teaching an introductory programming course using ADRI," Educ. Inf. Technol., vol. 22, no. 3, pp. 1089–1120, May 2017.

[4] G. Li, Y. Hou, and A. Wu, "Fourth Industrial Revolution: technological drivers, impacts and coping methods," Chinese Geogr. Sci., vol. 27, no. 4, pp. 626–637, Aug. 2017.

[5] S. Höme, J. Grützner, T. Hadlich, … C. D., and U. 2015, "Semantic industry: challenges for computerized information processing in Industrie 4.0. Automatisierung Stechnik," degruyter.com, vol. 63, no. 2, pp. 74–86, 2015.

[6] E. Sutherland, "The Fourth Industrial Revolution–The Case of South Africa," Politikon, vol. 47, no. 2, pp. 233–252, Apr. 2020.

[7] C. Tsekeris, "Surviving and thriving in the Fourth Industrial Revolution: Digital skills for education and society," Homo Virtualis, vol. 2, no. 1, p. 34, Mar. 2019.

[8] S. Xinogalos, M. Satratzemi, A. Chatzigeorgiou, and D. Tsompanoudi, "Factors Affecting Students' Performance in Distributed Pair Programming," J. Educ. Comput. Res., vol. 57, no. 2, pp. 513–544, Apr. 2019.

[9] G. Kanaparan, R. Cullen, and D. Mason, "Effect of Self-efficacy and Emotional Engagement on Introductory Programming Students," Australas. J. Inf. Syst., vol. 23, Jul. 2019.

[10] N. A. Bowman, L. Jarratt, K. C. Culver, and A. M. Segre, "How prior programming experience affects students' pair programming experiences and outcomes," in Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE, 2019, pp. 170–175.

[11] Simon et al., "Pass rates in introductory programming and in other STEM disciplines," in Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE, 2019, pp. 53–71.

[12] O. W. Adejo and T. Connolly, "Predicting student academic performance using multi-model heterogeneous ensemble approach," J. Appl. Res. High. Educ., vol. 10, no. 1, pp. 61–75, 2018.

[13] Z. Li, C. Shang, and Q. Shen, "Fuzzy-clustering embedded regression for predicting student academic performance," in 2016 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2016, 2016, pp. 344–351.

[14] D. Miler, M. M. Perišić, R. Mašović, and D. Žeželj, "Predicting student academic performance in Machine elements course," in Mechanisms and Machine Science, vol. 73, Springer Netherlands, 2019, pp. 825–834.

[15] C. Beaulac and J. S. Rosenthal, "Predicting University Students' Academic Success and Major Using Random Forests," Res. High. Educ., vol. 60, no. 7, pp. 1048–1064, Nov. 2019.

[16] P. J. A. C. van der Zanden, E. Denessen, A. H. N. Cillessen, and P. C. Meijer, "Domains and predictors of first-year student success: A systematic review," Educational Research Review, vol. 23. Elsevier Ltd, pp. 57–77, 01-Feb-2018.

[17] A. Alhadabi and A. C. Karpinski, "Grit, self-efficacy, achievement orientation goals, and academic performance in University students," Int. J. Adolesc. Youth, vol. 25, no. 1, pp. 519–535, Dec. 2020.

[18] G. Strong, C. Higgins, N. Bresnihan, and R. Millwood, "A survey of the prior programming experience of undergraduate computing and engineering students in Ireland," in IFIP Advances in Information and Communication Technology, 2017, vol. 515, pp. 473–483.

[19] Y. Qian and J. D. Lehman, "Correlates of Success in Introductory Programming: A Study with Middle School Students," J. Educ. Learn., vol. 5, no. 2, pp. 73–83, 2016.

[20] S. O. Akinola, O. A. Abiola, and E. Y. Ajao, "A Study on the Relationship among Background Knowledge , Non-Intellectual Strength and Programming Performance Using Empirical Analysis and Data Mining Technique," Int. J. Comput. Sci. Inf. Secur., vol. 17, no. 9, pp. 103–114, 2019.

[21] M. R. Bin Razak and N. Z. B. Ismail, "Influence of mathematics in programming subject," in AIP Conference Proceedings, 2018, vol. 1974, no. 1, p. 050011.

[22] H. Danielsiek and J. Vahrenhold, "Stay on these roads: Potential factors indicating students' performance in a CS2 course," in SIGCSE 2016 - Proceedings of the 47th ACM Technical Symposium on Computing Science Education, 2016, pp. 12–17.

[23] C. Stadtfeld, A. Vörös, T. Elmer, Z. Boda, and I. J. Raabe, "Integration in emerging social networks explains academic failure and success," Proc. Natl. Acad. Sci. U. S. A., vol. 116, no. 3, pp. 792–797, Jan. 2019.

[24] J. A. Craft, "Academic performance of low SES students at an Australian university satellite campus," Stud. High. Educ., vol. 44, no. 8, pp. 1372–1385, Aug. 2019.

[25] "Psychosocial Factors Influencing Academic Persistence of American Indian College Students," Journal of College Student Development. [Online]. Available: https://asu.pure.elsevier.com/en/publications/psychosocial-factors-influencing-academic-persistence-of-american. [Accessed: 08-Jun-2020].

[26] K. Talsma, B. Schüz, and K. Norris, "Miscalibration of self-efficacy and academic performance: Self-efficacy ≠ self-fulfilling prophecy," Learn. Individ. Differ., vol. 69, pp. 182–195, Jan. 2019.

[27] C. Y. Tsai, "Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy," Comput. Human Behav., vol. 95, pp. 224–232, Jun. 2019.

[28] G. Kanaparan, R. Cullen, and D. Mason, "Effect of Self-efficacy and Emotional Engagement on Introductory Programming Students," Australas. J. Inf. Syst., vol. 23, pp. 1–24, Jul. 2019.

[29] K. Ismayilova and R. M. Klassen, "Research and teaching self-efficacy of university faculty: Relations with job satisfaction," Int. J. Educ. Res., vol. 98, pp. 55–66, Jan. 2019.

[30] E. Aivaloglou and F. Hermans, "Early programming education and career orientation: The effects of gender, self-efficacy, motivation and stereotypes," in SIGCSE 2019 - proceedings of the 50th ACM echnical symposium on computer science education, 2019, pp. 679–685.

[31] J. Sweller, J. J. G. van Merriënboer, and F. Paas, "Cognitive Architecture and Instructional Design: 20 Years Later," Educational Psychology Review, vol. 31, no. 2. Springer New York LLC, pp. 261–292, 15-Jun-2019.

[32] R. S. Chauhan, "Unstructured interviews: are they really all that bad?," Hum. Resour. Dev. Int., pp. 1–14, 2019.

[33] C. Robson, "Real World Research," in Real World Research, 2nd ed., Blackwell Publishing Ltd, 2002.

[34] A. Carbonaro, "Good practices to influence engagement and learning outcomes on a traditional introductory programming course," Interact. Learn. Environ., vol. 27, no. 7, pp. 919–926, 2019.

[35] M. Te Wang and J. L. Degol, "School Climate: a Review of the Construct, Measurement, and Impact on Student Outcomes," Educational Psychology Review, vol. 28, no. 2. Springer New York LLC, pp. 315–352, 01-Jun-2016.

[36] M. P. Hewitt, "Self-Efficacy, Self-Evaluation, and Music Performance of Secondary-Level Band Students," J. Res. Music Educ., vol. 63, no. 3, pp. 298–313, Oct. 2015.

[37] A. Mehmood, M. Adnan, A. Shahzad, and F. Shabbir, "The Effect of Self-Efficacy on Academic Performance at Higher Level of Learning : A Case Study of Punjab University Lahore," J. Educ. Sci. Res., vol. 6, no. 1, pp. 33–46, 2019.