Ukrainian Journal
of Educational Studies and
Information Technology

ISSN: 2521-1234 online
Ukr. J. of Educ. Stud. and Inf. Technol.
11(2), 85 – 98
doi: 10.32919/uesit.2023.02.02

Vol. 11, Issue 2, 2023

The effects of students' perception on self-efficacy in programming courses at the tertiary institution

Isaac Ampofo Atta Senior

Submitted:
2022 – 02 – 17
Accepted:
2023 – 06 – 28
Published online:
2023 – 06 – 30

Abstract
Programming is a difficult subject to learn and teach. When it comes to students learning basic programming information and skills, university-level introductory programming courses (Java, C++, Visual Basic, and Python) are critical. Students' achievement is negatively impacted by a negative attitude about programming. As a result, the study discovered the impact of students' perceptions in university computer programming courses. The study covered students studying Computer Science from the University of Ghana. A survey descriptive design with a quantitative technique was used in this investigation. The population of the study was 2,030 with 368 sample size. Purposive sampling was utilized to choose University of Ghana, Legon as the study's location. The study's participants were chosen using a stratified random sampling technique. Closed-ended questionnaire was used for data collection. The SPSS version 26 and PROCESS Macro were used to analyze the data. Respondents' data were examined applying both inferential and descriptive statistics. The study revealed that students see programming as unfamiliar was the highest perception of programming to students. The study found that students see programming as easy with dedication was the lowest perception of programming to students. In conclusion, the significant impact of perception of students in Computer Programming account for 84% of the contribution of factors that influence self-efficacy.
Keywords: computer programming; self-efficacy; student perception; tertiary institution and course.

INTRODUCTION

Programming is a difficult subject to learn and teach. When it comes to students learning basic programming information and skills, university-level introductory programming courses (Java, C++, Visual Basic, and Python) are critical (Pears et al., 2007). Inappropriately, they likewise have the greatest rates of drop-out, and we've seen that even after passing beginning programming courses, students lack the necessary knowledge and abilities (Ampofo, 2021). It's worth noting that in the literature, introductory programming courses are sometimes "hidden" under the "Computer science" label (Radenski, 2006), making literature study extra challenging. The Computer Science Department at the University of Ghana's Faculty of Science (FOS) is in charge of teaching the majority of courses in programming for students majoring in mathematics, computer science, information technology, technical science, and physics, with many students, particularly in introductory courses

(Ampofo, 2021). Although there are not many students in comparison to additional campuses worldwide, it is notable in terms of the faculty. Because computer programming is seen as a challenging topic, students develop predominantly unfavorable views toward programming (Başer, 2013).

Students' achievement is negatively impacted by a negative attitude about programming. As a result, the study discovered the impact of students' perceptions in university computer programming courses (Korkmaz & Altun, 2013; Hawi, 2010; Anastasiadou & Karakos, 2011). It can be claimed that, in addition to programming courses, students' self-efficacy perceptions have a part in their performance in computer programming classes. Self-efficacy is an individual's belief in her or his own capability to do a task (Horzum & Akr, 2009). It is likely that you will come across research demonstrating that students with poor self-efficacy are more likely to fail a programming course (Aşkar & Davenport, 2009; Altun & Mazman, 2012). Demirtas, Cömert, and Özer (2011) found that these students' perceptions of programming differed depending on their class level in the literature. There is a small amount of research in the literature involving self-efficacy, programming, trust, and programming perception. There is a significant association between attitudes toward programming and exam results (Hongwarittorrn & Krairit 2010).

Students' attitudes toward programming are generally good, according to studies (Başer, 2013; Anastasiadou & Karakos, 2011; Nurazian, Haslzatul, Suzana, & Isamassabah, 2007; Korkmaz & Altun, 2013). Furthermore, while some research concludes that male students' attitudes toward programming are superior to female students. Others conclude that there is no link between gender and programming attitudes (McDowell, Fernald, Bullock, & Werner, 2003; Lau & Yuen, 2009). There has been no research on the impact of programming courses on students' perceptions of programming. However, there is little research on students' perceptions of their own programming self-efficacy. Perception of self-efficacy for programming differs significantly by department and gender, with computer engineering students and males having a greater perception of self-efficacy for programming than other students (Aşkar & Davenport, 2009). Programming self-efficacy is determined by the number of preceding courses taken and the end-of-year mark obtained from these courses. However, experience year has no influence (Jegede 2009). The perception of self-efficacy in programming does not differ by gender; however, it does vary by programming course and year of experience (Altun & Mazman 2012).

Furthermore, studies stating that students' perceptions of self-efficacy for programming are at an average level may be found in the literature (Robins, Rountree, & Rountree, 2003; Hawi, 2010; Pereira, Moret, & Zebende, 2010). Existing studies on programming attitudes and self-efficacy density have been conducted in computer and engineering education departments. There is research in the literature concerning pupil satisfaction from various departments in general, and specifically students from programming computers who take education (Ozyurt, 2014; Allen, Burrell, Bourhis, & Mabry, 2002), as well as student attitudes toward their distance education or/and department (Ojo & Olakulehin, 2006; Ozyurt, 2014; Ural, 2007; Lenka & Kant, 2012). There has never been a study that looked into the perception of students in Computer Programming and their impact on self-efficacy. In this context, this study's purpose is to evaluate computer programming candidates' perceptions of programming self-sufficiency. The study's content and scope are

expected to fill a significant vacuum in the literature and contribute to students' perceptions of programming.

LITERATURE REVIEW

Perception can be part into two strategies, handling the sensory input, which alters this low-level information to increasingly significant level data; training which is associated with an individual's ideas and desires (or knowledge), restorative and particular components, (for example, attention) that impact perception (Wagner, 2019). Perception depends upon multifaceted components of the sensory system, anyway passionately appears to be generally easy because this getting ready happens outside cognizant mindfulness (Koch, 2019). Some experts believe that students' mathematics talents have a good impact on their programming ability (Bennedsen, 2008). As a result, teachers prefer to create curricula that benefits these students, whereas other aspects for example solving problem may be overlooked (Sauter, 1986). Skilled programmers know more than beginners, nonetheless academics point out that knowledge quantity isn't the sole difference; specialists likewise arrange their information well (Lister, Thompson, Simon, Prasad, & Whalley, 2006). There is a tendency of beginner programmers to write specific-context programs and display only a rudimentary understanding of programming fundamentals (Lahtinen, Ala-Mutka, & Järvinen, 2005).

They understand grammar and small chunks of code, nonetheless they lack the capability to put together the program as a whole. "Learning strategy," "lack of study," "lack of practice," "teaching method," "exam anxiety," and "cheating" were among the ten factors identified by researchers in (Hawi, 2010), and we recognized some of these through observation and conversations with our students. Some of these criteria were also identified as indicators of programming effectiveness in (Bennedsen, 2008). Students face additional challenges when it comes to abstract thinking. (Eckerdal, Thun, & Berglund, 2005) conducted a study in which students were interviewed to see whether they know what studying programming entailed. Numerous students remarked that it is a unique thinking way, nonetheless were unable to elaborate. In the following part, we study students’ success rate at the FOS over 3 academic years to see whether there is a link between second- and first-semester beginning programming courses and mathematics introductory courses, as well as the success rate of subsequent programming courses.

Miqdadi and Harris conducted research into students' perceptions of first-year engineering tutorials (2019). The accompanying proposals, based on the study's findings, could help first-year students better understand instructional exercises. Teaching assistants (TAs) have enhanced their preparation to aid in improvement. Step-by-step directions for preparing for the educational activity and increasing proactivity to aid in the discovery of solutions to students' questions. Giving constructive feedback at the right time so that students can learn from it. Learning the importance of their vibrancy and demeanor in the classroom, as well as the extent to which it influences students' support and learning. The format of the instructional exercise has been altered, and students are now more likely to be found in groups.

While going through class material, the class is instructed as a whole. Later, the class is divided into groups to further analyze the topic and participate in exercises.

TAs catch up with groups, and students are given the ability to walk toward TAs on their own if necessary. Increase the number of in-class discussions and the use of internet applications. In the instructive exercise, practice more test questions. Increase TA availability or provide instructional exercise time for students to ask questions and receive feedback on assignments from TAs.

Blanchard, Gardner-McCune, and Anthony (2019) conducted research on code representation effects on pupil perceptions and programming attitudes. Many software engineering students regard content dialects as challenging, scary, and even boring. On the other hand, while squares-based settings are popular, many students believe they are unauthentic. Bidirectional half breed circumstances produce written and squares-based depictions of the same code, allowing students to seamlessly transition between depictions and develop a theoretical extension between squares and content. Notwithstanding, it is not realized how utilization of mixture conditions impacts the view of programming. To research, we led a study in an open centre school with six classes (n=129). We found that students who utilized cross breed situations saw message more decidedly than the individuals who moved straightforwardly from squares to content. The consequences of this exploration propose that crossover programming situations can assist with progressing students from squares to content-based programming while at the same time limiting the negative impression of programming.

A study by Abdunabi, Hbaci, & Ku (2019) on how to improve students' self-efficacy perceptions in programming in information systems. Right now, the Information Systems (IS) department in business colleges is moving towards incorporating figuring out how to program in their undergrad core courses. Numerous elements influencing IS student achievement in figuring out how to do programming remain distinguished with students' perception information deficiency all alone skill. This study research into what influence IS students' achievements in figuring out how to do programming. Students' attitude about worth as well as challenges to get the hang of programming can influence their abilities securing.

IS instructors need to comprehend the student perception identified with challenges of figuring out how to program to offer progressively viable help during their showing procedure and associations with learners. The investigation in other to curtail the issue inspects two basic components in enhancing educating IS customizing courses: (a) Students' programming Self-Efficacy convictions all alone programming capability, joined by (b) programming aptitudes levels which IS learners at first idea learn for future calling. The investigation uses quantitative information obtained from college students in Information Computer Systems classes at University of Colorado State in the U.S.A. what's more, bolstered by subjective information. Quantitative information quantifies the relationship among learners' programming self-adequacy, apparent benefit in programming, personal time for training and recurrence of teaching assistant (TA) discussions.

Qualitative data is used to comprehend learners' considerations of programming aptitudes for future calling that can impact the programming self-viability throughout the learning procedure. This research significance relies on the abilities of discoveries in the examination as basic, the most powerful influences that are probably going to be a medium through which instructors will either develop the personal viability of students or comprehend it all the more completely. Besides, these discoveries may

impact academic practices for showing programming dialects in advanced education settings all the more effectively. For example, executing a relevant learning method helps with recognizing the best way to deal with show programming courses, and thus, will prompt expanded learning results as experienced and described by computer science learners.

This correlation study showed significant positive connection concerning computer science students' programming self-adequacy and apparent benefit in getting the hang of programming. Be that as it may, the occurrence and training time of TA conferences required no huge association with program design self-adequacy. What's more, the subjective information uncovered a reasonable position of computer science learners' prospect vision of their five coding stages in all program design abilities: beginner, end-client, communicator, the expert than another classification with "equipped" rising. This writing allows computer science students to implant intercessions to expand their apparent benefit in getting training time in program design. It’s likewise extremely compelling in connecting real-life projects to class activities. Moreover, suggestions for instructors to use relevant studying method to bolster more elevated significant levels for program design and self-adequacy between IS students.

Additionally, harmonization among teachers and managers helps create compelling program design classes to improve computer science students' activity attractiveness. Researching different components that conceivably add to computer science students' customizing self-viability, for example, math introduction and past computer program design, inspiration, and monetary position. Understanding the significance of self-viability in program design can aid successful computer science instructors and proficient program courses to bring about students' getting the hang of program design with great success but few problems. Featuring the significance of connecting market wants with course substance can build learners' customizing self-adequacy with the odds of getting employment. Intelligent program design apparatus is a suggested addition for computer science teachers that build learner enthusiasm during training time to support learners' own chip away to appreciate the class and skilled instructors to precisely follow and survey learners' involvement.

METHODOLOGY

A design of research is a plan that demonstrates how to tackle an issue that is being investigated (Lewis, Chandler, Lawrence, & Colombo, 2019). A survey descriptive design with a quantitative technique was used in this investigation. Furthermore, the study aims to describe the characteristics of all students included in the study's population. Students at the University of Ghana, Legon, who were studying computer science were included in the study.

POPULATION

The term "population" refers to the group or persons that are being surveyed. It's the group of people to whom a researcher wishes to apply the findings of a study (Muianga, Barbutiu, & Hansson, 2019; Fischer, Karl, & Fischer, 2019). The study's respondents were students studying Computer Science from the University of Ghana.

Two thousand and thirty (2,030) students studying Computer Science from the University of Ghana, Legon were included in the study.

A SAMPLE AND SAMPLING TECHNIQUE

Purposive sampling was utilized to choose University of Ghana, Legon as the study's location. The University of Ghana, Legon was picked since it is one of West Africa's best universities. Three hundred and sixty-eight people were included in the study (368). The study's participants were chosen using a stratified random sampling technique (Celestine & Nonyelum, 2018). Because the researcher is outlining the entire population, stratified random sample correctly reveals the studied population (Murphy, 2020). The researcher chose stratified random sampling because it ensures that each segment within the population is represented appropriately in the sample (Murphy, 2020). A stratified sample ensured that the researcher had enough sample emphases to conduct a distinct population study.

DATA COLLECTION

The data for the examination of the subject under consideration was gathered using a questionnaire. Respondents were given a closed-ended questionnaire. The instructions were written in simple words so that respondents could grasp them. From "strongly disagree to strongly agree, a five-point Likert-type scale ranging from 1 to 5" was used (Mohamed, 2019). The researcher was given advance notice of when they would need their response. If they needed more information, the researcher provided them a phone number to call. The first of the research questions requested whether or not they learned programming as their course part. Participants were asked whether they wanted to participate or not, and if they said yes, they were then requested to fill the questionnaire. The research questionnaire was designed so that respondents may complete it in 15 minutes (Liao, Robert, Gurung, & Shi, 2015). The respondents' anonymity was safeguarded, and their provided information was given the strictest confidentiality (Gajjar, 2013). The study included suitable references and citations.

DATA ANALYSIS

The data was changed by carefully reviewing it to find any errors or questions that were incorrectly answered or responded to. The Statistical Package for Social Sciences (SPSS) version 22 and PROCESS Macro were used to analyze the data (Baran, Tondeur, Sari, & Bilici, 2019). This software was used since it has a user-friendly interface, stability, correctness, and the most used data analysis package. Respondents' data were examined applying both inferential and descriptive statistics (Adams, 2018; Guerin et al., 2019). To measure students' perception in programming, it was coded as SP.

RESULTS

Table 1 presents the results of items used to measure students' perception in programming. Table 1 demonstrated that I see programming as unfamiliar (SP5) had the highest mean of 3.2283 which means that, students see programming as unfamiliar was the highest perception of programming to students. The average distance a score was from the mean was 1.07337, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. I see programming as lovely when resources are easily accessible (SP2) had the second-highest mean of 3.2255 which means that students see programming as lovely when resources are easily accessible was the second-highest perception on programming to students. The average distance a score was from the mean was 1.48405, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed.

I see programming as difficult (SP6) had the third-highest mean of 3.1223, which means that students' perception of programming was the third-highest perception of programming to students. The average distance a score was from the mean was 1.04855, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. I see programming languages as being difficult to understand (SP8) had the fourth highest mean of 2.9293, which means that students' perception of programming was the fourth-highest. The average distance a score was from the mean was 0.88918, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed.

I see the difficulty in learning visual/script programming (SP12) had the fifth-highest mean of 2.8451 which means that students see the difficulty in learning visual/script programming was the fifth-highest perception on programming to students. The average distance a score was from the mean was 1.10507, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. I see the difficulty in learning procedural programming (SP11) had the sixth-highest mean of 2.6957 which means that students see the difficulty in learning procedural programming was the sixth-highest perception of programming to students. The average distance a score was from the mean was 1.07221, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed.

I see programming as time-consuming (SP3) had the seventh-highest mean of 2.5299 which means that students see programming as time-consuming was the seventh-highest perception of programming to students. The average distance a score was from the mean was 1.12885, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. I see the difficulty with concepts involving mathematical logic (SP10) had the eighth highest mean of 2.4212 which means that students see the difficulty with concepts involving mathematical logic was the eighth highest perception on programming to students. The average distance a score was from the mean was 1.08207, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed.

I see programming as entrepreneurial (SP4) had the ninth highest mean of 2.0082 which means that students see programming as entrepreneurial was the ninth highest

perception of programming to students. The average distance a score was from the mean was 0.76890, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. I see graphical programming languages to be helpful to young learners (SP9) had the tenth highest mean of 2.0082 which means that students see graphical programming languages to be helpful to young learners was the tenth highest perception on programming to students. The average distance a score was from the mean was 0.82034, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed.

I see programming as easy with dedication (SP7) had the eleventh highest mean of 1.8342 which means that students see programming as easy with dedication was the eleventh highest perception of programming to students. The average distance a score was from the mean was 0.83664, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. I see programming as easy with the help of tutorials (SP1) had the twelfth highest mean of 1.8071 which means that students see programming as easy with the help of tutorials was the twelfth highest perception on programming to students. The average distance a score was from the mean was 0.83405, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. The skewness was from -0.518 to 0.857 meaning that, the variable was normal sufficiently. Eleven items' kurtosis were less than 0 meaning that, it had fewer outliers comparative to normal distribution. Also, the kurtosis of one item was greater than 0 and less than 3 meaning that, it had few relatively outliers and scores were more clustered around the mean.

Table 1. **Descriptive statistics for students' perception**

	N	Minimum	Maximum	Mean	Std. Deviation	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
SP1	368	1.00	5.00	1.8071	.83405	.857	.127	.292	.254
SP2	368	1.00	5.00	3.2255	1.48405	.075	.127	-1.623	.254
SP3	368	1.00	5.00	2.5299	1.12885	.360	.127	-1.138	.254
SP4	368	1.00	4.00	2.0082	.76890	.167	.127	-.838	.254
SP5	368	1.00	5.00	3.2283	1.07337	-.518	.127	-.783	.254
SP6	368	1.00	5.00	3.1223	1.04855	-.446	.127	-.750	.254
SP7	368	1.00	5.00	1.8342	.83664	.629	.127	-.371	.254
SP8	368	1.00	5.00	2.9293	.88918	-.376	.127	-.295	.254
SP9	368	1.00	4.00	2.0082	.82034	.551	.127	-.142	.254
SP10	368	1.00	5.00	2.4212	1.08207	.321	.127	-1.005	.254
SP11	368	1.00	5.00	2.6957	1.07221	-.063	.127	-1.131	.254
SP12	368	1.00	5.00	2.8451	1.10507	.225	.127	-.809	.254

SP refers to students' perception in programming
Source: Researcher's fieldwork, (2020)

The impact of perception of students in Computer Programming on self-efficacy

The R value in Table 2 is .919[a] which means that the relationship between perception of students in Computer Programming and self-efficacy is strong and positive. The R Square of the model summary illustration is .844 which means the significant impact of perception of students in Computer Programming account for only 84% of the contribution of factors that influence self-efficacy. Hence, the model fit the study. The Std. Error of the Estimate between the variables is .42258 which is the average error for the model fit. How small the Std. Error of the Estimate is means that the model is good. The F Change for the model is 75.600 which is significant and it means that the R-squared does not equal to zero.

Table 2. **Model summary on impact of perception of students in Computer Programming on self-efficacy**

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	R Square Change	F Change	Sig. F Change
1	.919[a]	.844	.833	.42258	.844	75.600	.000

a. Predictors: (Constant), perception of students in Computer Programming

Hence, the relationship between self-efficacy and the model is statistically significant. Moreover, the p-value in Table 3 is .000[b] which is less than the F value of 75.600. This evidence that the regression model fits the data better than the model with no independent variables. The degree of freedom (367 – df1) is 366 which refers to the maximum number of logically independent values and have the freedom to vary in data sample without breaking any constraints.

Table 3. **ANOVA on impact of perception of students in Computer Programming on self-efficacy**

Model		df	F	Sig.
1	Regression	1	75.600	.000[b]
	Residual	366		
	Total	367		

a. Dependent Variable: self-efficacy

b. Predictors: (Constant), perception of students in Computer Programming

In Table 4, perception of students in Computer Programming showed a positive relation with on self-efficacy (B = .919, t = 8.695).

Table 4. **Coefficients on impact of perception of students in Computer Programming on self-efficacy**

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	-.500	.361		-1.386	.188
	perception of students in Computer Programming	1.500	.173	.919	8.695	.000

a. Dependent Variable: self-efficacy

DISCUSSION OF FINDINGS

The results of the study on students' perception of programming revealed that students' perception of programming as unfamiliar was the highest among students' programming perceptions. This finding is similar to that of Wagner (2019). Students see programming as lovely when resources are easily accessible. This is the second-highest perception of programming among students. Students' perception of programming as difficult was the third-highest perception of programming among students. Students' perception of programming as being difficult to understand was the fourth-highest perception of programming among students. Students' perception of the difficulty of learning visual/script programming was the fifth-highest perception of programming to students. Students' perception of the difficulty of learning procedural programming was the sixth-highest perception of programming to students. Students' perception of programming as time-consuming was the seventh-highest perception of programming among students. Students' perception of the difficulty with concepts involving mathematical logic was the eighth highest perception of programming among students. Students' perception of programming as entrepreneurial was the ninth highest perception of programming among students. Students' perception of graphical programming languages as being helpful to young learners was the tenth highest perception of programming among students. Students' perception of programming as easy with dedication was the eleventh highest perception of programming among students (Abdunabi, Hbaci, & Ku, 2019). Students' perception of programming as easy with the help of tutorials was the last perception of programming among students (Wagner, 2019). The relationship between self-efficacy and the model is statistically significant. Moreover, the p-value in Table 3 is $.000^{b}$ which is less than the F value of 75.600. This evidence that the regression model fits the data better than the model with no independent variables.

CONCLUSION

The significant impact of perception of students in Computer Programming account for only 84% of the contribution of factors that influence self-efficacy. Hence, the model fit the study and the relationship between self-efficacy and the model is statistically significant. In conclusion perception of students in Computer Programming showed a positive relation with on self-efficacy (B = .919, t = 8.695).

RECOMMENDATION

The study recommends that since the impact of perception of students in Computer Programming account for only 84% of the contribution of factors that influence self-efficacy, educational institutions, lecturers, curriculum designers, instructional designers and academia should look into the factors that can positively influence students' perception towards the learning of computer programming.

DECLARATION

Competing interests: The author declares no competing interests.

FUNDING

There was no external funding for the study.

REFERENCES

Abdunabi, R., Hbaci, I., & Ku, H.-Yu. (2019). Towards Enhancing Programming Self-Efficacy Perceptions Among Undergraduate Information Technology Students. *Journal of Information Technology Education: Research*, *18*, 185–206. DOI: https://doi.org/10.28945/4308.

Adams, D. R. (2018). *An Empirical Study on Teachers' and Students' Perception of Project Based Learning*. (Master's Thesis). University of Central Oklahoma. Retrieved from http://scholarworks.uark.edu/etd/2764.

Allen, M., Bourhis, J., Burrell, N., & Mabry, E. (2002). Comparing student satisfaction with distance education to traditional classrooms in higher education: A meta-analysis. *American Journal of Distance Education*, *16*(2), 83-88. DOI: https://doi.org/10.1207/S15389286AJDE1602_3.

Altun, A. & Mazman, S. G. (2012). Programlamaya ilişkin öz-yeterlilik algısı ölçeğinin Türkçe formumun geçerlilik ve güvenirlik çalişmasi. *Journal of Measurement and Evaluation in Education and Psychology*, *3*(2), 297–308. Retrieved from https://dergipark.org.tr/en/pub/epod/issue/5802/77220.

Ampofo, I. A. S. (2021). Mediating Effect of Students' Perception in Programming on the Relationship Between Project Work and College Students' Interest in Programming. *Research Square*, 1–14. DOI: https://doi.org/10.21203/rs.3.rs-935163/v1.

Anastasiadou, S. D. & Karakos, A. S. (2011). The beliefs of electrical and computer engineering students' regarding computer programming. *The International Journal of Technology, Knowledge and Society*, *7*(1), 37-52. DOI: https://doi.org/10.18848/1832-3669/CGP/v07i01/56170.

Askar, P. & Davenport, D. (2009). An investigation of factors related to self-efficacy for Java programming among engineering students. *The Turkish Online Journal of Educational Technology*, *8*(1), Article 3. Retrieved from http://www.tojet.net/articles/v8i1/813.pdf.

Baran, E., Bilici, S. C., Sari, A. A., & Tondeur, J. (2019). Investigating the impact of teacher education strategies on preservice teachers' TPACK. *British Journal of Educational Technology*, *50*(1), 357–370. DOI: https://doi.org/10.1111/bjet.12565.

Baser, M. (2013). Developing attitude scale toward computer programming. *The Journal of Academic Social Science Studies*, *6*(6), 199-215. DOI: http://doi.org/10.9761/JASSS1702.

Bennedsen, J. (2008). Teaching and learning introductory programming: a model-based approach. (Doctoral dissertation). University of Oslo, Oslo. Retrieved from https://www.duo.uio.no/bitstream/handle/10852/9962/bennedsen.pdf.

Blanchard, J., Gardner-McCune, C., & Anthony, L. (2019). Effects of Code Representation on Student Perceptions and Attitudes Toward Programming. In *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, Memphis, TN, USA, 14-18 October 2019 (pp. 127–131). DOI: https://doi.org/10.1109/VLHCC.2019.8818762.

Celestine, A. U. & Nonyelum, O. F. (2018). Impact Of Social Media On Students' Academic Performance. *International Journal of Scientific & Engineering Research*, *9*(3), 1454–1462. Retrieved from https://www.ijser.org/researchpaper/IMPACT-OF-SOCIAL-MEDIA-ON-STUDENTS-ACADEMIC-PERFORMANCE.pdf.

Chang, S.-L., Shieh, R. S., Liu, E. Z. F., &. Yu, P.-T. (2012). Factors influencing women's attitudes regarding computers in a computer literacy training program. *Turkish Online Journal of Educational Technology*, 11(4), 177-187. Retrieved from http://www.tojet.net/articles/v11i4/11417.pdf.

Demirtas, H., Cömert, M., & Özer, N. (2011). Pre-service teachers' self-efficacy beliefs and attitudes regarding profession. *Education and Science*, *36*(159), 96-111. Retrieved from https://avesis.inonu.edu.tr/yayin/51bed9a9-7a0e-469b-b2cb-31a39700f32c/pre-service-teachers-self-efficacy-beliefs-and-attitudes-towards-profession.

Eckerdal, A., Thun, M., & Berglund, A. (2005). What does it take to learn 'programming thinking'? In *Proceedings of the first international workshop on Computing education research (ICER '05)*. (pp. 135–142). Association for Computing Machinery, New York, NY, USA. DOI: https://doi.org/10.1145/1089786.108979.

Fischer, R., Karl, J. A., & Fischer, M. V. (2019). Norms Across Cultures : A Cross-Cultural Meta-Analysis of Norms Effects in the Theory of Planned Behavior. *Journal of Cross-Cultural Psychology*, *50*(10), 1112–1126. DOI: https://doi.org/10.1177/0022022119846409.

Gajjar, N. B. (2013). Ethical Consideration in Research. *International Journal for Research in Education*, *2*(7), 1–8. Retrieved from http://www.raijmr.com/ijre/wp-content/uploads/2017/11/IJRE_2013_vol02_issue_07_02.pdf.

Guerin, R. J., Toland, M. D., Okun, A. H., Rojas-Guyler, L., Baker, D. S., & Bernard, A. L. (2019). Using a Modified Theory of Planned Behavior to Examine Teachers' Intention to Implement. *Journal of School Health*, *89*(7), 549–559. DOI: https://doi.org/10.1111/josh.12781.

Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computers & Education*, *54*(4), 1127-1136. DOI: https://doi.org/10.1016/j.compedu.2009.10.020.

Hongwarittorrn, N. & Krairit, D. (2010). Effects of program visualization (Jeliot3) on students' performance and attitudes regarding java programming. In *The spring 8th International conference on Computing, Communication and Control Technologies*, 6-9 April, Orlando, Florida USA. Retrieved from http://iiis.org/CDs2010/CD2010IMC/CCCT_2010/PapersPdf/TA750PM.pdf.

Horzum, M. B. & Çakir, Ö. (2009). The validity and reliability study of the Turkish version of the online technologies self-efficacy scale. *Educational Sciences: Theory & Practice*, *9*(3), 1345-1356. Retrieved from https://files.eric.ed.gov/fulltext/EJ858927.pdf.

Jegede, P. O. (2009). Predictors of Java programming self-efficacy among engineering students in a Nigerian University. *International Journal of Computer Science and Information Security*, *4*(1&2), 08070905. Retrieved from https://arxiv.org/abs/0909.0074.

Koch, C. (2019). *Student Perceptions Of Reasons For Lecture And Active Learning*. (Undergraduate Honors Theses). University of Nebraska–Lincoln. Retrieved from http://digitalcommons.unl.edu/honorstheses/72.

Korkmaz, O. & Altun, H. (2013). Engineering and CEIT student's attitude regarding learning computer programming. *The Journal of Academic Social Science Studies*, *6*(2), 1169-1185. DOI: http://doi.org/10.9761/jasss_690.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, *37*(3), 14-18. DOI: https://doi.org/10.1145/1067445.1067453.

Lau, W. W. F. & Yuen, A. H. K. (2009). Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy. *British Journal of Educational Technology*, *40*(4), 696-712. DOI: https://doi.org/10.1111/j.1467-8535.2008.00847.x.

Lenka, S. K. & Kant, R. (2012). A study of attitude and perception of the learners regarding distance education in relation to their biographical factors. *Turkish Online Journal of Distance Education*, *13*(4), 236-244. Retrieved from https://dergipark.org.tr/en/pub/tojde/issue/16902/176212.

Lewis, K. O., Colombo, J., Lawrence, C., McNeal-Trice, K., & Chandler, M. (2019). "Making Learning and Teaching Fun Again!" – Strategies to Improve Learner Engagement and Retention. *Presentations*, 15. Retrieved from https://scholarlyexchange.childrensmercy.org/presentations/15.

Liao, Q., Luo, X. (R.), Gurung, A., & Shi, W. (2015). A holistic understanding of non-users' adoption of university campus wireless network : An empirical investigation. *Computers In Human Behavior*, *49*, 220–229. DOI: https://doi.org/10.1016/j.chb.2015.02.044.

Lister, R., Simon, B., Thompson, E., Whalley, J. L., & Prasad, C. (2006). Not seeing the forest for the trees: novice programmers and the SOLO taxonomy. *ACM SIGCSE Bulletin*, *38*(3), 118-122. DOI: https://doi.org/10.1145/1140124.1140157.

McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2003). The impact of pair programming on student performance, perception and persistence. In *25th International Conference on Software Engineering, 2003. Proceedings.*, Portland, OR, USA, 2003 (pp. 602-607). DOI: https://doi.org/10.1109/ICSE.2003.1201243.

Miqdadi, M. & Harris, J. (2019). Investigating Students' Perceptions Of First – Year Engineering Tutorials. In *Proceedings of the Canadian Engineering Education Association (CEEA)*. DOI: https://doi.org/10.24908/pceea.vi0.13786.

Mohamed, H. (2019). *Effect of Mobile Banking on the Financial Performance of Commercial Banks in Kenya*. United States International University-Africa.

Muianga, X. J., Barbutiu, S. M., Hansson, H., & Mutimucuio, I. V. (2019). Teachers' perspectives on professional development in the use of SCL approaches and ICT: A quantitative case study of Eduardo Mondlane University, Mozambique. *International Journal of Education and Development Using Information and Communication Technology*, *15*(2), 79–97. Retrieved from http://ijedict.dec.uwi.edu/viewissue.php?id=54.

Murphy, C. B. (2020). Stratified Random Sampling: Advantages and Disadvantages. *Investopedia*. Retrieved from https://www.investopedia.com/ask/answers/041615/what-are-advantages-and-disadvantages-stratified-random-sampling.asp.

Nurazian, M. D., Suzana, B., Haslizatul, F. M. H., & Ismassabah, I. (2007). Development of instruments for measuring learning attitudes regarding computer programming. In *National Conference on Programming Science (ATUR 07)*, 5 December 2007, Pacific KLIA, Selangor.

Ojo, D. O. & Olakulehin, F. K. (2006). Attitudes and perceptions of students to open and distance learning in Nigeria. *International Review of Research in Open & Distance Learning*, *7(*1). DOI: https://doi.org/10.19173/irrodl.v7i1.313.

Ozyurt, H. (2014). Satisfaction clustering analysis of distance education computer programming students: A sample of Karadeniz Technical University. *Turkish Online Journal of Distance Education*, *15*(2), 53-61. Retrieved from https://dergipark.org.tr/en/pub/tojde/issue/16892/175946.

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M., & Paterson, J. (2007). A survey of literature on the teaching of introductory programming. *ACM SIGCSE Bulletin*, *39*(4), 204-223. DOI: https://doi.org/10.1145/1345443.1345441.

Pereira, H. B. de B., Zebende, G. F., & Moret, M. A. (2010). Learning computer programming: Implementing a fractal in a Turing Machine. *Computers & Education*, *55*(2), 767-776. DOI: https://doi.org/10.1016/j.compedu.2010.03.009.

Radenski, A. (2006). "Python first": a lab-based digital introduction to computer science. *ACM SIGCSE Bulletin*, *38*(3), 197-201. DOI: https://doi.org/10.1145/1140123.1140177.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, *13*(2), 137-172. DOI: https://doi.org/10.1076/csed.13.2.137.14200.

Sauter, V. L. (1986). Predicting computer programming skill. *Computers & Education*, *10*(2), 299-302. DOI: https://doi.org/10.1016/0360-1315(86)90031-X.

Stoilescu, D. & Egodawatte, G. (2010). Gender differences in the use of computers, programming, and peer interactions in computer science classrooms. *Computer Science Education*, *20*(4), 283-300. DOI: https://doi.org/10.1080/08993408.2010.527691.

Ural, O. (2007). Attitudes of graduate students toward distance education, educational technologies and independent learning. *Turkish Online Journal of Distance Education*, *8*(4), 34-43. Retrieved from https://dergipark.org.tr/en/pub/tojde/issue/16922/176602.

Wagner, K. M. (2019). *Faculty and Students' Perceptions of Open Educational Resources vs. Traditional Textbooks*. (Doctoral Thesis). Wilmington University.

About the author:

Isaac Ampofo Atta Senior, Akenten Appiah-Menka University of Skills Training and Entrepreneurial Development, Kumasi, Ghana. ORCID: https://orcid.org/0000-0003-4279-4403. ampofoisaac10@yahoo.com