

Mediating Effect of Students' Perception in Programming on the Relationship Between Project Work and College Students' Interest in Programming

Isaac Ampofo Atta Senior (✉ ampofoisaac10@yahoo.com)

University of Education Winneba

Research Article

Keywords: Computer programming, project work, student, interest, perception

Posted Date: September 27th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-935163/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Abstract

Computer programming has been a subject of debate whether students have interest or not. Students' impressions of their own abilities to program and of programming in general are influenced by their early programming experiences, which play a role in encouraging or discouraging students' interest in computing areas. As a result, project work as a method of teaching has been adopted by many lecturers in teaching computer programming. However, the study sought to assess the mediating influence of students' perception in the relationship between project work and college students' interest in programming. The study used a descriptive survey design. The target population for the study included all Computer Science and Mathematics students in the University of Cape Coast, Ghana. The population for the study was 2,410 students with a sample size of 332. The researcher used a questionnaire to gather the data for studying the issue under investigation. Analysis of the data was done through the use of the PROCESS Macro. The results revealed that the mediating effect of students' perception in programming on the relationship between project work and college students' interest in programming account for only 0.34% of the contribution of factors that influence college students' interest in programming.

Introduction

Text-based production languages (i.e., Python, C/C++, and Java) have typically been used in programming teaching (Tharp, 1982). This has the advantage of anchoring education in industry-standard languages (Disalvo, 2014), but it poses challenges for new computer science students. When compared to other types of programming environments, working with text has been demonstrated to increase students' cognitive burden (Bau et. al. 2017). In focus group interviews, inexperienced students expressed their perceptions of text-based programming languages as difficult, frightening, and/or tedious (Begel & Klopfer, 2007). Students' impressions of their own abilities to program (Biggers, Brauer, & Yilmaz, 2008) and of programming in general are influenced by their early programming experiences, which play a role in encouraging or discouraging students' interest in computing areas (Disalvo, 2014). Many educators employ project work programming environments to expose students to programming without the difficulties of text-based programming, especially in the case of younger pupils (Weintrop & Wilensky, 2016; Goode & Margolis, 2011). The simplicity with which students can learn to program in a project-based environment has aided in the introduction of programming to a wide number of students in higher institutions (Cooper, Dann & Pausch, 2000; Maloney, Resnick & Rusk, 2010; Goschnick, 2015). While many students believe project work oriented programming is easier to use than text, they also believe it is more expensive and time consuming (Disalvo, 2014; Weintrop & Wilensky, 2015). Furthermore, it has been noted that some students find project-based programming challenging to master. However, instructors and researchers in programming have indicated a need for project work based programming to help students understand how to program for the industry, as well as to find solutions to overcome misconceptions about project work based programming's complexity (Bau et. al. 2017). Some members of the computer science education community believe that project-based programming increases the

likelihood of attaining desired outcomes. As a result, the mediating influence of students' perception in the relationship between project work and college students' interest in programming is investigated in this study.

Studying Computer Programming

Benefits during instructional delivery of computers accept that the genuine estimation of computers lies in the capacity of people to instruct computers to perform a specific function that individual will take more time before he/she can execute (Al-Mohammadi, 2015). The best way to guide a computer to do the necessary tasks is by programming it orders like human language, called programming language. Languages in programming is a middle person between a client and a computer. Therefore, there is a need to incorporate computer programming in the educational programs of the state-funded education (Alhassan & Arabia, 2017). There are normal qualities and likenesses between most present-day programming languages. The overall aptitudes of programming of the parts of comparability are program preparation, putting down algorithms, drafting flowcharts, managing numbers and factors, managing restrictive sentences, managing loops, and dealing with arrays and functions (Waziri, 2014).

One might say that considering programming skills of any programming language (Visual Basic Studio, Scratch, and Java.) is to become familiar with the skills of programming by and large, yet through explicit language. Along these lines, a student or a developer can master programming aptitudes in a specific language and when the person wants to gain proficiency with a later language, the learning procedure will be simpler than learning full programming skills as a result of the indistinguishable fundamental programming skills in most current languages (Lee, 2014). Concerning the techniques for surveying students getting the hang of programming skills, there are two fundamental strategies for appraisal in the field of programming (Mohammed, Metwally, & Ali, 2015). First, evaluating the intellectual side, the side handles the data and information for learning programming and it is measured by diverse sorts of tests accomplishment. Next is evaluating the specialized side, this side surveys the students' capacity to write programming code within the necessary speed and exactness. This perspective is regularly surveyed by requesting that a student write a product code, to actualize and to check its legitimacy and results.

Student Engagement Theory

Student engagement theory uses limited cognizance of condition as a great deal of students' conditions affected on students' behaviour towards programming. We envision that such importance of institutional environment confines our comprehension into inward components of student commitment. We cannot clarify why there is a partition in the responsibility of students, who learn in one class and have the same condition, by these theories (Maloshonok, 2014). The theory of student engagement in learning programming accentuates the role of the condition in the learning of students. Schools should bolster student engagement in programming by presenting convincing academic practices throughout the learning technique. The insightful outcomes and improvement of students are influenced by student

commitment in learning programming. Engagement theory relies upon making compelling shared groups that work on goal-oriented activities that are imperative to an individual outside the study room.

These three fragments, consolidated by Relate-Create-Donate, gather that learning works out happen in cooperative groups are venture based and have a valid focus (Shneiderman & Kearsley, 2018). Regardless of the way that student engagement with the academic work of programming is basic to students' accomplishment and their social and scholarly improvement, commitment deduces both full of feeling and conduct support in the learning experience (Marks, 2000). Engagement in programming lesson is characterized as a powerful arrangement of social and mental grows similarly as a synergistic system (Lawson & Lawson, 2013). Moreover, in numerous investigations, engagement and disengagement are seen and evaluated on alone continuum, with lower levels of commitment (Fredricks, 2016). So students' engagement in teaching and learning of programming is crucial as it influence their perception towards the course to know how to program and correct error codes.

Self-explanation strategy on obtaining computer programming skills and its effect with worked examples by Alhassan & Arabia (2017). The reason for this research was to inspect the impact of using self-clarification learning procedure bolstered with examples that are worked on gaining computer programming abilities among rookie's secondary school students. This research embraced a semi trial technique, where an experimental group (n = 33) utilized the self-clarification procedure upheld with worked models in getting the hang of programming, and a control group (n = 31) figured out how to do programming using the learning strategy characterized in the National Guidelines for teaching computer curriculum. The consequences of the research demonstrated that students in the exploratory group accomplished fundamentally better in programming knowledge and abilities contrasted with the control group. This research prescribed to remember the self-explanation methodology for computer programming courses in secondary school computer education reading material.

Numerous past studies have discovered a constructive outcome of the use of the self-explanation technique on the students' learning in various topics (Lee, 2014; Kwon & Jonassen, 2011; Yuasa, 1994; Bielaczyc, 1995). The discoveries of the present studies are by these discoveries in the field of computer programming learning, both the information and abilities. The consequences of the present study are reliable with the outcome of various old studies led by Pirolli & Recker, (1994), as the outcomes demonstrated a beneficial outcome for the use of self-explanation procedure upheld with worked models when learning subjects identified with programming. This beneficial outcome was evident using the students' capacity to compose the right programming code and articulations, and the capacity to discover mistakes in given programming. Similarly significant, the present study demonstrated a positive effect of supporting the self-explanation methodology by using worked models that make it simpler for students to learn new programming ideas, which might be seen as a confusing issue by various new students.

This positive effect for the combination of these two procedures could be ascribed to their role in decreasing the psychological exertion of the student and deliberately directing that person to the most

ideal approach to tackle the programming issues. The self-explanation methodology adds to the students' learning by testing the constraints of their thoroughly considering posing many inquiries about what is being instructed (Lee, 2014). Further conceivable support for these positive discoveries in accomplishment in programming information and abilities maybe because of the use of self-clarification technique when learning programing as this system assists with understanding complex ideas all the more profoundly. Various studies demonstrated that controlling students to use self-explanation technique encourages them to see a new topic with a more noteworthy profundity of learning than that on account of a conventional technique for instructing, particularly when the system is bolstered with worked models (Roy & Chi, 2005; McNamara, 2009).

The self-clarification methodology may likewise assist students with arriving at positive outcomes as far as data recovery and the use of the past skills in new teaching settings, and make students progressively independent in the development of significance through their studies of the slip-ups and rectifying them, which may prompt the maintenance of gained ideas and aptitudes for a more extended time (Lee, 2014; Kalyuga, 2009; Van-Merriënboer & Sluijsmans, 2009). Furnishing students with worked models causes them, beat the normal missteps submitted by new students in computer programing. Such models offer the right use of the programming orders in numerous specific circumstances, which empowers them to compose new programming code with insignificant mistakes (Kwon & Jonassen, 2011). Furthermore, when learning computer programing, specifically, which includes a psychological weight (Cognitive Load), the use of worked models may help soothe this psychological over-burden (Garner, 2002).

Using the Storytelling Alice programming environment to create computer-animated movies inspires middle school girls' interest in learning to program computers by Pausch & Kelleher (2007). In looking for approaches to make computer science all the more speaking to students, instructors must consider more than just the quantities of students they enlist. While making educational programs dependent on games, they should be certain that the games they pick truly can rouse students of the two genders. However, games like the Sims that intrigue unequivocally to the two sexual orientations is excessively mind boggling for first-semester students. As we scan for games that are of sensible degree for starting students in a game world that is not impartial, we risk picking games, (for example, first-individual shooters) that excessively advance to male students. Except if we take care in the manner we plan our gaming educational programs, exhibiting wide intrigue among the bigger gaming community, we risk additionally decreased decent variety in computer science. How might we be able to perhaps think of it as a triumph on the off chance that we multiplied the number of students studying computer science while further diminishing the level of computer science majors who are ladies?

Methodology

A research design is a design that shows how a problem under investigation can be solved (Lewis, Colombo, Lawrence, & Chandler, 2019). The study used a descriptive survey design. The study used a survey strategy because not much is known about the study and the research questions of the study use 'what'. Also, the study intends to describe the characteristics of all students used as a population in the

study. A quantitative research study was used by the researcher (Walton, Homan, Naimi, & Tomovic, 2008). The design enabled the researcher to select a sample which constituted respondents from the different working environment to reflect those of the larger population. The descriptive research enabled the researcher to carefully construct standardized questionnaires that provided data in the same form from all respondents (Grace-martin, Gay, & Ph, 2014). The research design was appropriate since the researcher described some aspects of a population by selecting unbiased samples of programming students at the University of Cape Coast, Ghana who was asked to complete the questionnaire.

Population

The study was conducted at the University of Cape Coast, Ghana because the researcher wanted the study to be done in the Central Region. Central Region is one of the sixteen regions in Ghana. The population of the study was Computer Science and Mathematics students. The target population for the study included all Computer Science and Mathematics students in the University of Cape Coast, Ghana. The population for the study was two thousand four hundred and ten (2,410). There were nine hundred and thirty-five (935) students of Mathematics and one thousand four hundred and seventy-five (1,475) students of Computer Science.

Sample and Sampling Technique

The technique of purposive sampling was used in selecting University of Cape Coast, Ghana for the study. The selected university for the study is among the best universities in Ghana. In purposive sampling, the cases to be used in the sample are handpicked based on their judgment of their typicality or particularly knowledgeable about the issues under study (Trek, 2020). The researcher used an online sample size calculator from survey monkey with a population of two thousand four hundred and ten (2,410) with a confidence level of 95% and a margin error of 5%. A confidence level of 95% means the researcher is 95% sure that the findings for the sample will be the same as the findings for the population. The sample size was three hundred and thirty-two (332). For the sample size to be representative, the researcher used a stratified random sampling technique to select participants for the study (Celestine & Nonyelum, 2018).

Stratified random sampling precisely mirrors the population being examined because the researcher is delineating the whole population (Murphy, 2020). The total population for males were one thousand eight hundred and twenty-four (1,824) whiles five hundred and eighty-six (586) were females. Since there are plenty of males in the population, the majority of males were selected over females for the study. The researcher selected the sample size by calculating a fraction of males over the population ($1,824/2,410$) of the sample for men which was two hundred and fifty-one (251) and another fraction of females over the population ($586/2,410$) of the sample for females which was eighty-one (81). The researcher used stratified sampling in selecting the participants.

The researcher used stratified random sampling because it guarantees every subgroup inside the population gets appropriate portrayal within the sample (Murphy, 2020). As a result, stratified random

sampling gives better inclusion of the population since the researcher has authority over the subgroups to guarantee every one of them are represented in the sampling (Mohamed, 2019). A stratified sample can give more prominent accuracy so it frequently requires a smaller sample, which saves money (Trek, 2020). A stratified sample can make preparations for an unrepresentative sample like an all-male sample from a mixed-gender population (Trek, 2020). A stratified sample was guaranteed that the researcher got adequate sample focuses to help a separate analysis of the population.

Research Instrument and Measurement

The researcher used a questionnaire to gather the data for studying the issue under investigation (Asih, 2019; Lin et al., 2019). The questionnaire for respondents was close-ended (Ngitoria, 2014; Kataria, Krishna, Tyagi, & Vashishat, 2019). Simple sentences were used for the wording such that respondents could understand the instructions. Five-point Likert-type scale was given ranging from '1' to '5'- from 'strongly disagree' to 'strongly agree' (Yukselturk & Altioek, 2017; Mohamed, 2019). (Ma, Gam, & Banning, 2017). Although the sample size for the study was three hundred and thirty-two (332) but the researcher shared three hundred and fifty questionnaires to avoid any inconveniences. Four items were adapted from a study by Chen et al. (2018) to design the questionnaire and measure students' perception, another four items were adapted from a study by Chen et al. (2018) to measure students' interest in programming. Another four items were adapted from a study by White et al. (1997) to measure students' interest in programming.

Data Collection Procedure

Data were collected within one week. A letter of permission was sent to the Head of Departments of Computer Science and Mathematics at the University of Cape Coast, Ghana to permit the researcher to use the students for the research site. The data collection was done through Google forms. Prior notice was given to the participants as to when the researcher needed their response. The researcher gave them a telephone number to call for further explanation if needed. The first question in the research questions asked whether they learn programming as part of their course or not. Priorities were given as to whether participants want to take part or not, if 'Yes' then the participants proceeded on to answer the items in the questionnaire. The research questionnaire was created to take 15 minutes for the respondents to answer the items in it (Liao, Robert, Gurung, & Shi, 2015).

Data Analysis Procedure

Analysis of the data was done through the use of the Statistical Package for Social Sciences (SPSS) version 22 and PROCESS Macro (Baran, Bilici, Sari, & Tondeur, 2019). This software was selected because of its reliability, accuracy, user friendly interface and the most employed package for analysing data. Data collected from the respondents were analysed using inferential statistics (Guerin et al., 2019; Adams, 2018).

Ethical Consideration

In conducting the study, participation in the study was voluntary and respondents were informed about their rights to decline participation at any time. To obtain respondents' informed consent, an explanation about the reasons and purpose of the study was communicated to them (Jacques, St, Tribble, & Pierre, 2019). The anonymity of the respondents was ensured and the data that they provided was treated with the utmost confidentiality (Gajjar, 2013). Appropriate citations and references were done in the study.

Results

H1: There exists a direct significant effect of project work on students' perception in programming.

Table 7 illustrate the direct significant effects of project work on students' perception in programming. The R Square of the model summary illustration is .0304 which means the significant effects of project work account for only 3.04% of the contribution of factors that influence students' perception in programming. The significant value is .0008 which is below .01 and .05 with a coefficient value of .2776 and a standardized coefficient of .1744 (see Appendix B). This indicates that there was a direct significant effect of project work on students' perception in programming.

Table 1: Independent Variable – Project Work

Model Summary						
R	R-sq	MSE	F	df1	df2	P
.1744	.0304	.2135	11.4870	1.0000	366.0000	.0008

Source: Researcher's fieldwork, (2020)

H2: There exists a direct significant mediating effect of students' perception in programming on the relationship between project work and college students' interest in programming.

Table 8 illustrate the direct significant mediating effect of students' perception in programming on the relationship between project work and college students' interest in programming. The R Square of the model summary illustration is .0034 which means the mediating effect of students' perception in programming on the relationship between project work and college students' interest in programming account for only 0.34% of the contribution of factors that influence college students' interest in programming (see Appendix B). From the model in table 8, the significant value (p) for the mediating effect of students' perception in programming on the relationship between project work and college students' interest in programming is .2750 which is above .05 with a coefficient value of .0819 and a standardized coefficient of .0580. This indicates that there was no direct significant mediating effect of students' perception in programming on the relationship between project work and college students' interest in programming.

H3: *There exists a direct significant impact of students' perception in programming on college students' interest in programming.*

Also, from table 8, the significant value (p) for students' perception in programming on college students' interest in programming is .9806 which is above .05 with a coefficient value of .0011 and a standardized coefficient of .0013 (see Appendix B). This indicates that there is no statistically significant effect of students' perception in programming on college students' interest in programming.

Table 2: Mediating Role of Students' Perception

Model						
	coeff	se	t	p	LLCI	ULCI
Constant	2.3643	.1836	12.8789	.0000	2.0033	2.7253
Student Perception	.0819	.0749	1.0932	.2750	-.0654	.2292
Project Work	.0011	.0471	.0243	.9806	-.0914	.0937

Source: Researcher's fieldwork, (2020)

Discussion Of Findings

Also, the significant effects of project work account for only 3.04% of the contribution of factors that influence students' perception of programming. The results is similar to that of Mazaya (2019). The mediating effect of students' perception in programming on the relationship between project work and college students' interest in programming account for only 0.34% of the contribution of factors that influence college students' interest in programming. The results is similar to that of Wagner (2019) and Yukselturk & Altiok (2017). The significant effects of project work account for only 0.34% of the contribution of factors that influence college students' interest in programming. The results is similar to that of Miqdadi & Harris (2019).

Declarations

Competing interests: The author declares no competing interests.

References

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

Al-Mohammadi, N. (2015). The effectiveness of a proposed educational software on the achievement of a computer course of first-grade high school students in Jeddah. *Arab Studies in Education and Psychology - Saudi Arabia*, 62, 305–327.

- Alhassan, R., & Arabia, S. (2017). The Effect of Employing Self-Explanation Strategy with Worked Examples on Acquiring Computer Programming Skills. *Journal of Education and Practice*, 8(6), 186–196. Retrieved from <http://www.iiste.org>
- Asih, S. S. (2019). Learning Discipline Level and its Correlation with the Achievement Level of Learning Outcomes of Primary School Students. *Advances in Social Science, Education and Humanities Research. 5th International Conference on Education and Technology (ICET 2019)*, 382(Icet), 361–363. Retrieved from <http://creativecommons.org/licenses/by-nc/4.0/>
- 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. <https://doi.org/10.1111/bjet.12565>
- Bau, D., Gray, J., Kelleher, C., Sheldon, J. & Turbak, F. (2017). Learnable programming: blocks and beyond. *Commun. ACM*, 60(6), pp. 72–80.
- Begel, A. & Klopfer, E. (2007). StarLogo TNG: An introduction to game development. *J. E-Learning*, 53, p. 146.
- Bielaczyc, K., & Pirolli, P. (1995). Training in Self-explanation and Self-Regulation Strategies: Investigating the Effects of Knowledge Acquisition Activities on Problem Solving. *Cognition and Instruction*, 13(2), 221–252.
- Biggers, M., Brauer, A. & Yilmaz, T. (2008). Student perceptions of computer science: a retention study comparing graduating seniors with CS leavers. in *Proc. 39th SIGCSE technical symposium on Computer science education*, pp. 402–406.
- 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.
- Chen, C., Haduong, P., Brennan, K., Sonnert, G., Chen, C., Haduong, P., ... Sonnert, G. (2018). The effects of first programming language on college students' computing attitude and achievement: a comparison of graphical and textual languages. *Computer Science Education*, 00(00), 1–26. <https://doi.org/10.1080/08993408.2018.1547564>
- Cooper, S., Dann, W. & Pausch, R. (2000). Alice: a 3-D tool for introductory programming concepts. *Journal of Comput. Sci. Coll.*, 15(5), pp. 107–116.
- Disalvo, B. (2014). Graphical qualities of educational technology: Using drag-and-drop and text-based programs for introductory computer science. *IEEE comput. graph. appl.*, 34(6), pp. 12–15.
- Fredricks, J. A. (2016). Student engagement, context, and adjustment : Addressing definitional, measurement, and methodological issues. *European Association for Research on Learning and Instruction*, 43, 2014–2017. <https://doi.org/10.1016/j.learninstruc.2016.02.002>

- 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
- Garner, S. (2002). Reducing the Cognitive Load on Novice Programmers. *Paper presented at the Ed-Media 2002 World Conference on Educational Multimedia, Hypermedia & Telecommunications*.
- Goode, J. & Margolis, J. (2011). Exploring computer science: a case study of school reform," *ACM Trans. Comput. Educ.*, 11(2).
- Goschnick, S. (2015). App review: ScratchJr (Scratch Junior). *Int. J. People Oriented Program*, 4(1), pp. 50–55.
- Grace-martin, M., Gay, G., & Ph, D. (2014). Web Browsing, Mobile Computing and Academic Performance. Web Browsing, Mobile Computing and Academic Performance. *Educational Technology & Society*, (January 2001). Retrieved from <https://www.researchgate.net/publication/220374645>
- Guerin, R. J., Toland, M. D., Okun, A. H., Rojas-Guylar, 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. <https://doi.org/10.1111/josh.12781>
- Jacques, M. C., St, D., Tribble, C., & Pierre, J. (2019). Filters in the coping process of people with schizophrenia: A constructivist grounded theory study. *Willey Online Library Journal*, (February), 142–152. <https://doi.org/10.1111/jpm.12515>
- Kalyuga, S. (2009). Instructional designs for the development of transferable knowledge and skills: A cognitive load perspective. *Computers in Human Behavior*, 25(2), 332–338.
- Kataria, Y. S., Krishna, H. G., Tyagi, V. K., & Vashishat, T. (2019). Consumer Buying behavior of Organic food Products in India Through the Lens of Planned Behavior Theory ABSTRACT: *Research Journal of Humanities and Social Sciences*, 10(01), 60–67. <https://doi.org/10.5958/2321-5828.2019.00011.1>
- Kwon, K., & Jonassen, D. (2011). The Influence of Reflective Self-explanations on Problem-solving Performance. *Journal of Educational Computing Research*, 44(3), 247–263.
- Lawson, M. A., & Lawson, H. A. (2013). New Conceptual Frameworks for Student Engagement Research, Policy, and Practice. *Review of Educational Research*, 83(3), 432–479. <https://doi.org/10.3102/0034654313480891>
- Lee, L. (2014). *The Effect of self-explanation and Reading Questions and answers on Learning Computer Programing*. (Unpublished Doctoral Dissertation). University of Nevada [UMI number: 3613893].
- Lewis, K. O., Colombo, J., Lawrence, C., & Chandler, M. (2019). Strategies to Improve Learner Engagement and “ Making Learning and Teaching Fun Again !” – Strategies to Improve Learner Engagement and

Retention. *Presentations*, 15, 2–81. Retrieved from <https://scholarlyexchange.childrensmercy.org/presentations/15>

Liao, Q., Robert, X., Gurung, A., & Shi, W. (2015). Computers in Human Behavior A holistic understanding of non-users ' adoption of university campus wireless network: An empirical investigation. *COMPUTERS IN HUMAN BEHAVIOR*, 49, 220–229. <https://doi.org/10.1016/j.chb.2015.02.044>

Lin, C., Broström, A., Årestedt, K., Mårtensson, J., Steinke, E., Pakpour, A. H., ... Mårtensson, J. (2019). Using extended theory of planned behavior to determine factors associated with help-seeking behavior of sexual problems in women with heart failure: a longitudinal study. *Journal of Psychosomatic Obstetrics & Gynecology*, 0(0), 1–8. <https://doi.org/10.1080/0167482X.2019.1572743>

Ma, Y. J., Gam, H. J., & Banning, J. (2017). Perceived ease of use and usefulness of sustainability labels on apparel products: application of the technology acceptance model. *Fashion and Textiles*, 4(3), 1–20. <https://doi.org/10.1186/s40691-017-0093-1>

Maloney, J., Resnick, M. & Rusk, N. (2010). The Scratch programming language and environment. *ACM Trans. Comput. Educ.*, 10(4), pp. 1–15.

Maloshonok, N. (2014). Vygotsky ' s Theory: Lessons for Student Engagement Research. *SERU International Research Conference*, 2–13. Moscow: National Research University.

Marks, H. M. (2000). Student Engagement in Instructional Activity: Patterns in the Elementary, Middle, and High School Years. *American Educational Research Journal*, 37(1), 153–184.

Mazaya, M. S. (2019). Effective practical learning model for the subject of basic information technology Effective practical learning model for the subject of basic information technology. In J. of Physics (Ed.), *International Conference on Mathematics and Science Education (ICMScE 2018)*. <https://doi.org/10.1088/1742-6596/1157/4/042003>

McNamara, D. S. (2009). The Importance of Teaching Reading Strategies. *Perspectives on Language and Literacy*, 34–40.

Miqdadi, M., & Harris, J. (2019). Investigating Students ' Perceptions Of First -Year Engineering Tutorials. *Proceedings 2019 Canadian Engineering Education Association (CEEA-ACEG19) Conference*, 1–5.

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

Mohammed, M., Metwally, A., Ali, N. (2015). The effectiveness of cognitive trips via the Web in the development of programing skills for the third preparatory grade students. *Journal of the College of Education, Banha University - Egypt*, 26, (101), 235–263.

Murphy, C. B. (2020). Stratified Random Sampling: Advantages and Disadvantages. Retrieved August 14, 2020, from Investopedia website: <https://www.investopedia.com/ask/answers/041615/what-are-advantages-and-disadvantages-stratified-random-sampling.asp#:~:text=Stratified random sampling accurately reflects the population being studied because,proper representation within the sample.>

Ngitoria, D. J. (2014). *EFFECTIVENESS OF MICROFINANCE INSTITUTIONS IN EMPOWERING OF WOMEN PETTY TRADERS' PERFORMANCE: THE CASE OF PRIDE TANZANIA LTD – MOROGORO BRANCH*. Mzumbe University.

Pausch, R., & Kelleher, C. (2007). Using the Storytelling Alice programming environment to create computer-animated movies inspires middle school girls' interest in learning to program computers. *COMMUNICATIONS OF THE ACM*, 50(7), 58–64.

Pirolli, P., & Recker, M. M. (1994). Learning Strategies and Transfer in the Domain of Programming. *Cognition and Instruction*, 12(3), 235–275.

Roy, M., & Chi, M. T. H. (2005). The Self-explanation Principle in Multimedia Learning. In R. E. Mayer (Ed.), *The Cambridge Handbook of Multimedia Learning* (271–286). New York, NY: Cambridge University Press.

Shneiderman, B., & Kearsley, G. (2018). Engagement Theory: A Framework for Technology-Based Teaching and Learning. *Educational Technology Publications, Inc.*, 38(5), 20–23. Retrieved from <http://www.jstor.org/stable/44428478>

Tharp, A. L. (1982). Selecting the 'right' programming language. in *Proc. 13th SIGCSE technical symposium on Computer science education*. pp. 151–155.

Trek, S. (2020). Stratified Random Sampling. Retrieved August 14, 2020, from Stat Trek website: <https://stattrek.com/survey-research/stratified-sampling.aspx>

Van Merriënboer, J. J. G. & Sluijsmans, D. M. A. (2009). Toward a Synthesis of cognitive Load Theory, Four-Component Instructional Design, and Self-Directed Learning. *Educational Psychology Review*, 21(1), 55–66.

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

Walton, A., Homan, S., Naimi, L., & Tomovic, C. (2008). Student perceptions of a wireless audience response system. *Emerald Insight*, 5(4), 217–229. <https://doi.org/10.1108/17415650810930901>

Waziri H., Khadhran, N., & Mustafa, A. (2014). The effectiveness of a proposed multimedia interactive program using Flash software on the development of some programming skills of preparatory school students. *Reading and Knowledge Journal*. Egypt, 149, 117–138.

Weintrop, D. & Wilensky, U. (2015). To block or not to block, that is the question: students' perceptions of blocks-based programming. *in Proc. 14th international conference on interaction design and children*, pp. 199–208.

Weintrop, D. & Wilensky, U. (2016). Bringing blocks-based programming into high school computer science classrooms. *Paper presented at Annual meeting of the American Educational Research Association*.

White, G. L., Sivitanides, M. P., & Marcos, S. (1997). A Theory of the Relationships between Cognitive Requirements of Computer Programming Languages and Programmers' Cognitive Characteristics. *Journal of Information Systems Education*, 13(1), 59–66.

Yuasa, M. (1994). *The effects of active learning exercises on the acquisition of SQL query writing procedures*. Georgia Inst of Technology.

Yukselturk, E., & Altioek, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, 48(3), 789–801. <https://doi.org/10.1111/bjet.12453>

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [APPENDIXB.docx](#)