Indonesian Journal of Informatics Education

Research Paper ISSN: 2549-0389

The Influence of Project Work Approach on College Students' Interest in Programming at the Private Universities in Ghana

Isaac Atta Senior Ampofo
Department of Computer Science
Kwame Nkrumah University of Science and Technology
Kumasi, Ghana
ampofoisaac10@yahoo.com

Isaac Atta Junior Ampofo
Department of Human Resource and Organizational Development, KNUST School of Business, Kwame Nkrumah
University of Science and Technology Kumasi, Ghana
ampofoisaac159@yahoo.com

Abstract:

Failure rates have become a global problem as university students studying computer programming grow worldwide. Students' interest has been linked to learning skills that require metacognition and critical thinking, which are essential for studying computer programming efficiently. As a result, the project work approach in studying computer programming combines knowledge of technology with soft skills. Project work is best suited for complicated problem-solving tactics and teamwork creatively. The study used quantitative methodology and a descriptive design survey to evaluate the project work approach's influence on college students' interest in programming. The study's participants were Christian Service University computer science students. A total of 420 students were enrolled in the study, with a sample size of 368. Inferential and descriptive statistics were applied to analyze the data received from the respondents. The study found that standalone systems were the highest factor in the project work given to students. The study revealed that project work could make students interested in programming. The study concluded that project work has a favorable and considerable impact on college students' programming interests.

Keywords: Computer Programming, College Students, Programming Course, Project Work Approach, Students' Interest.

DOI: http://dx.doi.org/10.20961/ijie.v6i1.59461

Introduction

Failure rates have become a global problem as the number of university students studying computer programming grows worldwide (Bennedsen & Caspersen, 2019; Topalli & Cagiltay, 2018). As said by O'Kelly and Gibson (2006), this is also true in the case of first-year computer science students. According to Robins et al. (2003), programming courses, particularly, are widely regarded as challenging and frequently have the top rates of dropout/failure. The School Informatics of Costa Rica's Universidad Nacional is no exception. Between 2008 and 2012, the average failure rate (including dropouts) of basic course programming was 47.2 percent. This rise in failure rates has sparked curiosity about the factors influencing achievement in an introductory computer programming course. Wilson and Shrock (2001) investigated whether gender, math background, encouragement, previous experience in programming, work preference style, and course level comfort impact success. The findings revealed that the most significant impacting element was math background following comfort level. The researcher stressed the necessity of giving students a welcoming, non-threatening setting that encourages learning, emphasizing emotions' importance in education.

Keeping students motivated has never been easy (Ampofo, 2021). Jenkins (2001) looked at four motivation categories in undergraduate computer science students: intrinsic, extrinsic, accomplishment, and social motivation. The findings revealed that extrinsic motivation is substantial, implying that an enormous proportion of students are driven to learn computer programming since they consider it would provide them with benefits such as increased job possibilities. This research also revealed that a nearly equal number of students are intrinsically motivated, which means they are genuinely invested in learning to improve their skills. Furthermore, the author noted that, in general, students driven naturally appeared to be extra involved in studying instead of computer programming. This research revealed that understanding and motivating kids to learn computer programming is complicated. Adopting a project work approach in learning programming is a current trend to make computer science courses extra sensational and intriguing for students (Blank, 2006; Cliburn, 2006; Klassner & Anderson, 2003; Rouhani & Jørgensen, 2022).

Despite some contradictory findings (McNally et al., 2006; Fagin & Merkle, 2002), it is widely assumed that using a project-based learning method in programming offers students an engaging learning environment (McWhorter & O'Connor, 2009; Lukkarinen et al., 2021). Students' motivation has been linked to learning skills, for example, metacognition and critical thinking, which are essential for studying computer programming efficiently (Bergin et al., 2005; Benton et al., 2018; Younis et al., 2021). Furthermore, because project work approaches combine knowledge of technology with soft skills, for example, complicated problem-solving tactics and teamwork, they are best suited for stimulating the solving of problems creatively (Hees et al., 2011; Morais et al., 2021). Instead, some researchers have emphasized the importance of a project work approach (PWA) in motivating students and lowering rates of dropout and failure (Nuutila et al., 2005; Hamalainen, 2004; Lane, 2021). Through hands-on practice skills, collaborative skills, planning and project management skills, and problem-solving skills, PWA helps develop student learning (Pereira et al., 2021; Kolmos et al., 2004). PWA is thought to have a large influence on intrinsic motivation for studying since students can comprehend what they are studying for and why (Prince & Felder, 2006; Chen et al., 2021).

Furthermore, according to Prince and Felder (2007), the strategy encourages active and collaborative learning as well as increased student accountability for her or his own studying. In an introductory course of programming at the Christian Service University in Ghana, experimental research was conducted to further evaluate the effectiveness of employing PWA to minimize rates of dropout and failure and impact student motivation. Several studies have been done on using a project work approach to enhance teaching and learning (Nashirova, 2021; Shuhailo & Derkach, 2021; Tran et al., 2021; Sobral, 2021). Currently, no studies have been done on using the project work approach in teaching and learning computer programming to enhance students' interest. Also, the style of project work approach used in this study is new compared to previous literature (Sobral, 2021; Wei et al., 2021; Glazunova et al., 2021; López et al., 2021). The study contributes to the literature by looking at the use of the project work approach in teaching and learning at the university to enhance students' interest in computer programming. The project work approach in this study refers to teaching computer programming students through mini projects such that the lecturer will develop a program with the students as part of lecture periods. For example, in developing a game application, the lecturer will start the project as a form of teaching and complete the application by making sure that students have gotten the concepts before assigning them a similar game application. It is believed in this study that when this method of project work approach in teaching computer programming is enhanced at the university, it will arouse the intrinsic, extrinsic, accomplishment, and social motivation factors of college students to influence their interest positively. Given this, the study addresses the following research questions;

1. What type of software is used for project work at the university?
2. Can project work make college students interested in programming?
3. What is the effect of project work on college students' programming?

Related Work

The Project Work Strategy (PWA) is a pupil-based learning method in which students study by addressing an open-ended challenge (Ampofo, 2021). PWA is based on constructivist concepts, encourages active learning, and encourages collaboration (Hissey, 2000; ACM, 2013; Prince & Felder, 2006; Kanygin & Koretckaia, 2022). The approach attempts to mimic a scenario of work-centered, either in terms of problem discovery and definition or as a real-life project simulation with multiple ways to implement the answer or solve the problem. Instead of being a teacher, students work in minor groups with the teacher acting as a facilitator or supervisor. As students study to be extra autonomous in their method rather than depend solely on professors, the method has the ability to increase responsibility and motivation in the process of learning (Dirckinck-Holmfeld, 2002; Loyens et al., 2008; Bråting & Kilhamn, 2022). According to Prieto (2006), PWA encourages students to use real-world challenges as a springboard for learning and integrating new information. The strategy encourages the development of decision-making, problem-solving, communication abilities, and teamwork.

These traits are very beneficial in the field of engineering computers. In computer engineering, the capacity to resolve issues is crucial, and many of the actions of professionals are defined in projects' development. As a result, ACM (2011) specifies many abilities that prospective alums need to possess, including effective communication, problem-solving, professional responsibility, practical cooperation, and the ability to learn for the rest of their lives. Several studies in higher education have looked at the effectiveness of PWA versus lecture-based teaching. The outcomes are incongruent. According to Kinnunen and Malmi (2005), contingent on whether learning emphasis is on factual knowledge acquisition or on social skills, learning self-directed skills, and motivation, the findings favor the other or one. PWA may improve skill levels, but it may also cause students to do worse on standard test subjects and be stressful. As a result, lectures, exercises, and other pedagogical activities are included in some PWA learning designs. In a study involving the use of PWA in introductory programming courses, Nuutila et al. (2005) discovered an important reduction in the rates of dropout.

According to the authors, students learn programming, communication skills, independent study, and collaborative work. In a similar vein, Hamalainen (2004), the author of a study on using PWA to teach computer science theoretical concepts, claims that dropout and failure rates are lower when students follow a PWA rather than a traditional one. Furthermore, when compared to a regular course, the author states that students are more committed to the PWA course. There have also been some issues with employing the PWA technique. Both students and teachers will experience a cultural shift as a result of PWA. Students are accustomed to lecture-centered teaching approaches, which encourages them to adopt a passive attitude and cast the instructor in an expert role. Other issues are linked to PWA's core features: challenges as a learning stimulus, group work as an interaction stimulus, and tutors as facilitators (Dolmans et al., 2005). Students in some learning environments are presented with tasks that are overly regimented and closed. In this situation, the challenges are too simple to actively encourage students to develop knowledge. According to Dolmans et al. (2005), another factor that impedes the PWA learning process is an overbearing or lenient supervisor, leading to group tension and conflict, student absenteeism, and lack of dedication. Regarding group work, some groups are dysfunctional, exhibiting motivation and absence of cohesion, obstructing learning and collaborative nature (Dolmans et al., 2005; Kinnunen & Malmi, 2005; Chorfi et al., 2022). From the inference of literature, there is no study assessing the influence of project work approach on college students' interest in programming (Sobral, 2021; Wei et al., 2021; Glazunova et al., 2021; López et al., 2021). To bridge the literature gap, the study sought to examine the influence of the project work approach on college students' interest in programming at a private university. The study hypothesized that;

H1: *Project work significantly directly affects college students' programming interests.*

Research Method

The study used an experimental study for the project work approach and a descriptive design survey to evaluate the project work approach's influence on college students' interest in programming. A quantitative research study was applied (Walton et al., 2008). The study covered computer science students at the university. The study was done at various private universities in Ghana. Computer science students were given hands-on experience in all areas of software during the experimental research, including standalone systems, interactive transaction-based software, embedded systems, batch processing systems, systems for modeling and simulation, information systems, data collection systems, and entertainment systems. Over the last five years, computer science students have programmed various software. Following the experimental investigation, the researcher used a descriptive design to pick a sample of respondents from various working environments to represent the greater community. The descriptive research allowed the creation of uniform questionnaires that collected data from all respondents in a similar format.

Population and Sample Size

There are 73 universities owned by private individuals and organizations as of 2019. There are 64, 870 private university students representing 20.5% of the total university students in Ghana. Computer science students at the private universities represent 5% of 20.5%. As a result, the total population of computer science students at the private universities are 3,243. However, a total of 420 people were enrolled in the study where three hundred and twenty-four (324) represent males and ninety-six (96) represent females. Private universities in Ghana was chosen using the purposive selection technique because they are one of Ghana's best universities. The sample size for the study was 368, which was after the data collection. The study was done to cover the population as the sample size, but after the data collection, only 368 computer science students answered the questionnaire completely.

Data Collection

The data for the study's analysis was gathered via a questionnaire. Respondents were given a closed-ended questionnaire (Kataria et al., 2019). The instructions were written in simple words so that respondents could grasp them (Gajjar, 2013). From "strongly disagree to strongly agree, a five-point Likert-type scale ranging from 1 to 5" was used (Yukselturk & Altiok, 2017). Four items were adopted from Chen et. al. (2018)'s study to assess students' programming interest. The study added 4 items from research conducted by White et al. (1997) to assess students' programming interests. The rest of the questionnaire's items were created by the researcher. The questionnaire data collection took two weeks. Because of the worldwide Covid-19 pandemic, the researcher used Google forms to develop the questionnaire and gave the link to the participants through their WhatsApp group with instructions on how to complete it. They were given advance notice of when the researcher required their response. If they needed more information, the researcher provided them a phone number to call. The research questionnaire was designed so that respondents could complete it in 15 minutes (Liao et al., 2015).

Ten colleagues were given the information in the questionnaire to explain how they understood and would respond to the questions. Their replies were analyzed with Statistical Package for Social Sciences (SPSS) to see how well the information in the questionnaire measured the objectives and how closely it corresponded to the study's outcome (students' interest in programming). This was done to see if the information in the questionnaire corresponded to the study's aims. It was also important to know if the data in the questionnaire is representative. The reliability test result for Project Work Approach was.79, while the interest in programming among college students was.723. The researcher conducted the main survey with computer science students since the questionnaire was legitimate and dependable. The data was examined with the SPSS version 26. (Baran et al., 2019; Jacques et al., 2019). Inferential and descriptive statistics were applied to analyze the data received from the respondents (Adams, 2018; Guerin et al., 2019).

Result and Analysis

The descriptive results for the current study emphasized on the influence of project work on students' programming perception at the private universities in Ghana. It shows the mean, standard deviation, minimum and maximum Likert choice chosen by respondents, and the total number of respondents (Al-adwan, Box, & Arabia, 2016). The results also show the hypothesis that tests the effect and relationship between the variables.

The Type of Software Project Work Given to Students

Table 1 shows the descriptive statistics of responses to project work. Table 1 demonstrated that standalone systems had the highest mean of 2.4266, which means that standalone systems were the highest factor in the project work given to students. The average distance a score was from the mean was 0.84495, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. The entertainment system had the second-highest mean of 2.3234, which means that the entertainment system was the second highest factor in the project work given to students. The average distance a score was from the mean was 0.98285, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. The batch processing system had the third-highest mean of 2.2853, which means that the batch processing system was the third-highest factor of the project work given to students. The average distance a score was from the mean was 0.72900, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. The system for modelling and simulation had the fourth highest mean of 2.1223, which means that a system for modelling and simulation was the fourth-highest factor of project work given to students. The average distance a score was from the mean was 0.95040, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed.

The embedded system had the fifth-highest mean of 2.0734, which means that the embedded system was the fifth-highest factor of the project work given to students. The average distance a score was from the mean was 0.84657, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. Interactive transaction-based software had the sixth-highest mean of 2.0707, which means that interactive transaction-based software was the sixth-highest factor of project work given to students. The average distance a score was from the mean was 0.61799, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed.

The data collection system had the seventh-highest mean of 1.7989, which means that the data collection system was the seventh-highest factor of project work given to students. The average distance a score was from the mean was 0.71405, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. The information system had the last mean of 1.7255, which means that the information system was the last factor of the project work given to students. The average distance a score was from the mean was 0.59352, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed.

Table 1. Descriptive Statistics for Project Work

	N	Minimum	Maximum	Mean	Std. Deviation
	Statistic	Statistic	Statistic	Statistic	Statistic
Standalone systems	368	1.00	4.00	2.4266	.84495
Interactive transaction-based software	368	1.00	4.00	2.0707	.61799
Embedded system	368	1.00	5.00	2.0734	.84657
Batch processing system	368	1.00	5.00	2.2853	.72900
System for modelling and simulation	368	1.00	5.00	2.1223	.95040
Information system	368	1.00	3.00	1.7255	.59352
Data collection system	368	1.00	4.00	1.7989	.71405
Entertainment system	368	1.00	5.00	2.3234	.98285

College Students' Programming Interest and Project Work Approach

Table 2 demonstrated that project work can make students interested in programming, with the highest mean of 2.8668, which means that project work can make students interested in programming. The average distance a score was from the mean was 1.56117, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. Project work helps to disclose positive information about programming had the second-highest mean of 2.4810, which means that project work helps to disclose positive information about programming. The average distance a score was from the mean was 1.16706, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed.

Project work exposes students to designing algorithms and had the third-highest mean of 2.2418, which means that project work exposes students to designing an algorithm. The average distance a score was from the mean was 0.74460, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. How project work directs the students can lead to enthusiasm had the fourth highest mean of 2.0870, which means that how project work directs the students can lead to enthusiasm. The average distance a score was from the mean was 0.96987, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed.

Project work makes students learn had the fifth-highest mean of 2.0000, which means that project work makes students learn. The average distance a score was from the mean was 0.93377, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. Project work that gives information to students had the sixth-highest mean of 1.8832, which means that project work gives information to students. The average distance a score was from the mean was 0.92199, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. Project work influences students' perception had the seventh-highest mean of 1.8723 which means that project work influences students' perception.

The average distance a score was from the mean was 0.62421, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed. Project work helps in understanding the linguistic structure just as the rationale of the program does, which means that project work helps in understanding the linguistic structure just like the rationale of the program. The average distance a score was from the mean was 0.61514, which is a measure of dispersion (standard deviation) that indicates how broadly the distribution was distributed.

Table 2. Descriptive Statistics for Project Work and College Students' Interest in Programming

	N	Minimum	Maximum	Mean	Std. D. S.
	Stat.	Stat.	Stat.	Stat.	
Project work exposes students in the designing algorithm	368	1.00	4.00	2.2418	.74460
Project work helps in understanding the linguistic structure just as the rationale of the program	368	1.00	3.00	1.8261	.61514
Project work gives information to students	368	1.00	5.00	1.8832	.92199
Project work can make students find interest in programming	368	1.00	5.00	2.8668	1.56117
Project work helps to disclose positive information about programming	368	1.00	5.00	2.4810	1.16706

How project work directs the students can lead to enthusiasm	368	1.00	4.00	2.087 0	.96987
Project work influences students' perception	368	1.00	3.00	1.872 3	.62421
Project work makes students learn	368	1.00	5.00	2.000 0	.93377

Source: Researcher's fieldwork, (2021)

H1: *There exists a direct significant effect of project work on college students' programming interest.*

The direct substantial impacts of project work on college students' programming interest are shown in Tables 3, 4, and 5. The model summary's R Square (see Table 3) image is.383, indicating that significant project work effects account for just 38.3 percent of factors' contribution influencing college students' programming interest. This is calculated as.383 x 100%, which will give a result of 38.3%. Moreover, in Table 4, the significant value is .000 which is below .05 with a mean square of 24.314 and a residual value of .107. Table 5 depicts the association between the adoption of a project-based learning strategy in teaching computer programming and the interest of college students in programming. Table 5 demonstrates a positive connection with a t value of 15.082, a coefficient value of.019, and a standardized coefficient value of.619. This suggests that project work has a favorable and considerable impact on college students' programming interest.

Table 3. Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.619[a]	.383	.382	.32694
a. Predictors: (Constant), Project Work Approach				

Source: Researcher's fieldwork, (2021)

Table 4. ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regressio n	24.314	1	24.314	227.47 2	.00 0[b]
	Residual	39.121	366	.107		
	Total	63.435	367			
a. Dependent Variable: CollegeInterest						
b. Predictors: (Constant), Project Work Approach						

Source: Researcher's fieldwork, (2021)

Table 5. Coefficients[a]

Model		Unstandardized Coefficients		Standardize d Coefficients	t	Si g.
		B	Std. Error	Beta		
1	(Constant)	1.845	.049		37.44 1	.0 00
	Project Work Approach	.279	.019	.619	15.08 2	.0 00
a. Dependent Variable: CollegeInterest						

Source: Researcher's fieldwork, (2021)

Discussion

The findings reveal that students' emotional and physiological well-being (specifically responses of emotion) are influenced by the project work method, in addition to learning process active participation. The students found working on the project work to be enjoyable. The opportunity to work on a project was viewed by the students as an exciting, unique opportunity, and fun. They enjoyed interactively working with the programming and were inspired by the learning method of trial-and-error that was required for project work. They did, however, show irritation and demoralization once they had to make minor variations to the computer code on a regular basis. Insignificant programming details, they believed, drew attention away from additional principles of general programming. Students also felt insecure and unsure about their learning outcomes, owing to a perceived lack of theoretical knowledge to assist them via the process of trial-and-error. They also sensed a sense of urgency since they had to find a comfortable area to work whenever they were required to work on project work. All of these activities took up a lot of time and pulled them away from their programming work.

The project work is strong and there is a lot of interaction and teamwork. However, the students don't have the freedom to develop and outline the project's objective. Similarly, the nature of project work forces students to distribute tasks among themselves, limiting their experience with studying. Together, the limitations inherent in using a project work approach to develop real-world programming tasks are a difficult learning design obstacle. The activity had no clear plan or task arrangement, and most PWA students contributed continuously and dynamically, possibly engaging every pupil in the best possible way and helping them to have a broad experience in learning. In a similar spirit, the participatory learning technique in class gave the students a sense of security, and they felt ready for the project work. Furthermore, the students emphasized the learning process of trial-and-error as well as the common interplay among personal concerns, co-student experiences and ideas, and teacher guidance. The joint effort was difficult, but it was mostly perceived as engaging and enjoyable, i.e. emotionally gratifying.

The PWA provided a productive setting for exploratory learning, as expected based on previous research. However, under the current setup, the degree of freedom to design the challenge and organize the process of problem-solving might be increased, providing students with more challenges and skills. In comparison to the two other designs, the total intensity of the motivating force appeared to be much lower. In general, students delegated studying process structure to the professor, seeming content to have a professor who supervised the process of problem-solving and introduced programming principles. It gave them confidence in the study's results. Nonetheless, they valued the one small project work, were inspired by the genuine difficulties, and enjoyed working with additional students. While students were required to organize the task individually, they felt insecure and unprepared about group work and the process; questions such as “How do we tackle the problem?” were raised. “Can you tell me how we're going to organize the work?” “How are we going to collaborate?” The learning classroom design's strength is its ability to provide control and continuity over the outcome of learning.

The results of the study on the kind of project work been given to students revealed that standalone systems were the highest kind of project work given to students, entertainment system was the second-highest kind of project work given to students, batch processing system was the third-highest kind of project work given to students, a system for modelling and simulation was the fourth-highest kind of project work given to students, the embedded system was the fifth-highest kind of project work given to students, interactive transaction-based software was the sixth-highest kind of project work given to students, data collection system was the seventh-highest kind of project work given to students, and information system was the least kind of project work given to students. The results is similar to that of Pausch and Kelleher (2007), Appiah-kubi (2018), Miqdadi and Harris (2019), Mahnic (2014), Burga et al. (2020), Habók and Nagy (2016), Adams (2018), and Mazaya (2019). The results of the study on project work influence on students' programming perception revealed that project work can make students find programming interest. The finding is like that of Appiah-kubi (2018). Project work helps to disclose positive information about programming. The finding is like that of Adams (2018).

Project work exposes students to designing an algorithm. The results are similar to those of Burga et al. (2020). How project work directs the students can lead to enthusiasm. The finding is consistent with that of Mazaya (2019). Project work makes students learn, which is similar to a study by Adams (2018) who found the project to improve student participation in class. Appiah-kubi (2018) also found that students see that they are roused to learn by either customer-based or non-customer-based projects. Project work gives information to students. Project work influences students' perceptions, which is similar to that of Ciência et al. (2019), who found that students in the project-based learning course expressed increasingly positive perspectives related to the project in regards to the learning procedure and polish. Project work helps in understanding the linguistic structure just as the rationale of the program. The findings are comparable to those of Miqdadi and Harris (2019). In conclusion, the study revealed that there is a positive and significant effect of project work on college students' programming interest.

Conclusion

To summarize, each of the three learning styles has its own set of benefits and drawbacks. The PWA appears to be the most effective in terms of encouraging collaboration, interaction, and emotional well-being. Due to intrinsic restrictions in project assignments, practical challenges, and insecurity about the learning outcome, the project work approach is interesting and motivating, but sometimes frustrating. Traditional classroom lectures provide stability in terms of theoretical comprehension, but they can cause worry and anxiety owing to little or no experience with actively and cooperatively working. None of the learning designs are totally fulfilling if we desire happy, comfortable, delighted students who are also quiet and dynamic. It is also clear that working on projects motivates all students. However, in order for project work to become an effective tool for motivation, more theoretical knowledge about programming must be provided, as well as the project tasks and conditions under which the project work approach is used in the course. The concept of combining and tying theoretical lectures to actual practice was a success. The students should, however, understand how the lectures fit into the broader curriculum. Otherwise, this method of speaking might make us doubt whether we are learning what we must be learning. Students' emotional response to conventional lectures demonstrates the need of personal touch between lecturers and students. By selecting and arranging the theoretical themes, the lecturer provides security as well as interest and motivation. The PWA strategy proved to be the most motivating and engaging. According to the findings of the study, project work accounts for just 38.3 percent of factors contribution that impact college students' programming interest. As a result, future research can focus on the 61.7 percent of factors (100 percent-38.3 percent) that impact college students' programming interests.

Acknowledgement

I give thanks to Beatrice Ampofo for proofreading the work.

References

ACM. (2011). *Computer Science Curricula 2013*. ACM. Retrieved from http://ai.stanford.edu/users/sahami/CS2013/.

ACM. (2013). *Computer Science Curricula 2013.* Retrieved from http://ai.stanford.edu/users/sahami/CS2013/.

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-adwan, M. M., Box, P. O., & Arabia, S. (2016). Management Information Systems & Their Impact on Improving the Quality of Service at the Commercial Bank Customers. *International Journal of Business and Social Science*, *7*(6), 1–16. Retrieved from www.ijbssnet.com

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. Retrieved from https://doi.org/10.21203/rs.3.rs-935163/v1

Appiah-kubi, P. (2018). Multivariate Analysis of Students Perception on Teaching with Client Based and Non-Client Based Team Projects. *IJEP*, *8*(3), 93–103. Retrieved from http://www.i-jep.org

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

Bennedsen, J., & Caspersen, M. E. (2019). Failure rates in introductory programming: 12 years later. *ACM Inroads*, *10*(2), 30–36.

Benton, L., Saunders, P., Kalas, I., Hoyles, C., & Noss, R. (2018). Designing for learning mathematics through programming: A case study of pupils engaging with place value. *International Journal of Child-Computer Interaction*, *16*, 68–76. https://doi.org/10.1016/j.ijcci.2017.12.004

Bergin, S., Reilly, R., & Traynor, D. (2005). Examining the role of self-regulated learning on introductory programming performance. *Proceedings of the 2005 International Workshop on Computing Education Research*, 81–86.

Blank, D. (2006). Robots Make Computer Science Education Personal. *Communications of the ACM.*, *49*(12).

Bråting, K., & Kilhamn, C. (2022). The integration of programming in Swedish school mathematics: investigating elementary mathematics textbooks. *Scandinavian Journal of Educational Research*, *66*(4), 594–609. https://doi.org/10.1080/00313831.2021.1897879

Burga, R., Leblanc, J., & Rezania, D. (2020). Exploring Student Perceptions of Their Readiness for Project Work : Utilizing Social Cognitive Career Theory. *Project Management Journal*, *51*(2), 154–164. https://doi.org/10.1177/8756972819896697

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

Chen, H., Ding, G., Qin, S., & Zhang, J. (2021). A hyper-heuristic based ensemble genetic programming approach for stochastic resource constrained project scheduling problem. *Expert Systems with Applications*, *167*, 114174. https://doi.org/10.1016/j.eswa.2020.114174

Chorfi, A., Hedjazi, D., Aouag, S., & Boubiche, D. (2022). Problem-based collaborative learning groupware to improve computer programming skills. *Behaviour & Information Technology*, *41*(1), 139–158. https://doi.org/10.1080/0144929X.2020.1795263

Ciência, D. De, Moreira, R., & Figueiredo, E. (2019). Students Perception on the use of Project - Based Learning in Software Engineering Education. *In XXXIII Brazilian Symposium on Software Engineering*, 537–546. Retrieved from https://doi.org/10.1145/3350768.3352457

Cliburn, D. C. (2006). A CS0 course for the liberal arts. *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, 77–81.

Dirckinck-Holmfeld, L. (2002). Designing Virtual Learning Environments Based on Problem Oriented Project Pedagogy. In *Learning in Virtual Environments* (pp. 31–54). Frederiksberg C: Samfundslitteratur Press.

Dolmans, D., De Grave, W., Wolfhagen, I., & van der Vleuten, C. (2005). Problem-based learning: future challenges for educational practice and research. *Medical Education*, *39*, 732–741.

Fagin, B. S., & Merkle, L. (2002). Quantitative analysis of the effects of robots on introductory Computer Science education. *Journal on Educational Resources in Computing*, *2*(4), 1–18.

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. https://doi.org/10.1111/josh.12781

Habók, A., & Nagy, J. (2016). In - service teachers ' perceptions of project - based learning. *SpringerPlus*, *5*(83), 1–14. https://doi.org/10.1186/s40064-016-1725-4

Hamalainen (2004). Problem-based learning of theoretical computer science. In *Proceedings of the 34th Annual Conference on Frontiers in Education* (Vol. 3, pp. 20– 23).

Hees, F., Jeschke, S., Natho, N. & Pfeiffer, O. (2011). Developing a PBL-based rescue robotics course. In *Automation, Communication and Cybernetics in Science and Engineering 2009/2010* (pp. 231–240). Berlin Heidelberg: Springer.

Hissey, T. W. (2000). Enhanced Skills for Engineers. *Proceedings of the IEEE*, *88*(8).

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

Jenkins, T. (2001). The motivation of students of programming. *ACM SIGCSE Bulletin*, *33*(3), 53–56.

Kanygin, G., & Koretckaia, V. (2022). Analytical Coding: Performing Qualitative Data Analysis Based on Programming Principles. *Qualitative Report*, *26*(2).

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

Kinnunen, P., & Malmi, L. (2005). Problems in Problem-Based Learning – Experiences, Analysis and Lessons Learned on an Introductory Programming Course. *Informatics in Education,*, *4*(2), 193–214.

Klassner, F., & Anderson, S. D. (2003). LEGO Mindstorms: Not just for K-12 anymore. *IEEE Robotics and Automation Magazine*, *10*(2), 12–18.

Kolmos, A., Fink, F., & Krogh, L. (2004). The Aalborg Model - Problem-based and Project Organized Learning. *In A. Kolmos, F. Fink, & L. Krogh (Eds.), The Aalborg PBL model - Progress, Diversity and Challenges* (pp. 9–18). Aalborg: Aalborg University Press.

Lane, D. (2021). *Machine learning for kids: A project-based introduction to artificial intelligence*. No Starch Press.

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

Loyens, S. M., Joshua, M., & Rikers, R. M. (2008). Self-Directed Learning in Problem-Based Learning and its Relationships with Self-Regulated Learning. *Educational Psychology Review*, *20*(4), 411–427.

Lukkarinen, A., Malmi, L., & Haaranen, L. (2021). Event-driven programming in programming education: a mapping review. *ACM Transactions on Computing Education (TOCE)*, *21*(1), 1–31.

Mahnic, V. (2014). Teaching Scrum through Team-Project Work : Students ' Perceptions and Teacher ' s Observations Teaching Scrum through Team-Project Work : Students ' Perceptions and Teacher ' s Observations. *Research Gate Publication*, (6), 1–24.

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

McNally, M., Goldweber, M., Fagin, B. S., & Klassner, F. (2006). Do LEGO Mindstorms robots have a future in CS education. *ACM SIGCSE Bulletin*, *38*(1), 61–62.

McWhorter, W. I., & O'Connor, B. C. (2009). Do LEGO® Mindstorms® motivate students in CS1? *In Proceedings of the 40th ACM technical symposium on Computer science education (SIGCSE '09)* (pp. 438–442). New York, NY, USA.

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

Morais, P., Ferreira, M. J., & Veloso, B. (2021). Improving student engagement with Project-Based Learning: A case study in Software Engineering. *IEEE Revista Iberoamericana de Tecnologias Del Aprendizaje*, *16*(1), 21–28.

Nuutila, E., Törmä, S., & Malmi, L. (2005). Pbl and computer programming - the seven steps method with adaptations. *Computer Science Education*, *15*(2), 123–142.

O'Kelly, J., & Gibson, P. (2006). RoboCode & problem-based learning: a non-prescriptive approach to teaching programming. In *ITICSE '06 Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education* (pp. 217–221).

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.

Pereira, R., Couto, M., Ribeiro, F., Rua, R., Cunha, J., Fernandes, J. P., & Saraiva, J. (2021). Ranking programming languages by energy efficiency. *Science of Computer Programming*, *205*, 102609. https://doi.org/10.1016/j.scico.2021.102609

Prieto, L. (2006). Aprendizaje activo en el aula universitaria: el caso del aprendizaje basado en problemas. *Miscelánea Comillas: Revista de Ciencias Humanas Y Sociales*, *64*(124), 173–196.

Prince, M., & Felder, R. (2006). Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of Engineering Education*, *95*, 123–138.

Prince, M., & Felder, R. (2007). The Many Faces of Inductive Teaching and Learning. *Journal of College Science Teaching*, *36*(5).

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, *13*(2), 137–172.

Rouhani, M., & Jørgensen, V. (2022). In-Service Teachers' Attitude Towards Programming for All. In *Ludic, Co-design and Tools Supporting Smart Learning Ecosystems and Smart Education* (pp. 149-162). Springer, Singapore.

Topalli, D., & Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers & Education*, *120*, 64–74. https://doi.org/10.1016/j.compedu.2018.01.011

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

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.

Wilson, B.., & Shrock, S. (2001). Contributing to success in an introductory computer science course: A study of twelve factors. *ACM SIGCSE Bulletin*, *33*(1), 184–188.

Younis, A. A., Sunderraman, R., Metzler, M., & Bourgeois, A. G. (2021). Developing parallel programming and soft skills: A project based learning approach. *Journal of Parallel and Distributed Computing*, *158*, 151–163.

Yukselturk, E., & Altiok, 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