College Students' Interest in Programming

Isaac Ampofo Atta Senior (ampofoisaac10@yahoo.com)
University of Education Winneba https://orcid.org/0000-0003-4279-4403

Isaac Ampofo Atta Junior
Kwame Nkrumah University of Science and Technology

Research Article

Keywords: Programming practice, Age categories, Engineering students, Special interest, Gender attitude

Posted Date: December 6th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1099754/v1

Abstract

In today’s world, tertiary institutions provide Computer Science and Information Technology courses to students for them to have the opportunity to improve the thinking computational skills they will need to function in today's computation-dependent society. Moreover, pupils who think they can finish programming tasks successfully are likely more to become interested. Additionally, effective approaches to impart computer programming frequently rise pupils' interest in programming. As a result, the study sought to examine college students’ interest in programming. The research design for the study was a survey method. The study’s population comprise of Computer Science and Information Technology students at Kwame Nkrumah University of Science and Technology. The sample size was 368 where we used a systematic sampling technique to select participants for the study. The study shows that majority of college students have interest in programming. The results revealed that the total number of males who have interest in programming are greater than females and that the total number of males who study programming far outweigh the total number of females. It is recommended that females should be encouraged to study programming at the various institutions that provide programming courses to students.

Introduction

Papert Seymour popularized computational thought concept and emphasized its empowerment power, claiming that by giving offspring computers access, intelligent and a suitable simple language of programming, and peripheral devices able to produce on-line-real-time activity, we can offer them unparalleled power to carry out and innovate sensational ventures (Papert, 1980). Education of computational thinking in this vein, has the ability to change pupils from mere technology users to technology creators (Angeli, Fluck, Voogt, Cox, Webb, Zagami, & Malyn-Smith, 2016), allowing them to have a positive effect on people's lives and technical environment (Brennan & Resnick, 2012). Computational thinking education can help students develop their imagination (Voogt, Good, Fisser, Yadav, & Mishra 2015; Deschryver & Yadav, 2015), as well as inspire them to become technology producers through the development of programming objects (Angeli et al., 2016; Voogt, Good, Fisser, Yadav, & Mishra 2015).

According to the previous research, a good programming atmosphere will help students develop their creative self-efficacy (Burke, 2012). Thus, when students are more interested in an assignment, they are more confident in their ability to complete it, and they are more likely to begin working on it and finish it (Chiu & Klassen, 2010). Pupils who think they can finish a programming job successfully are likely more to become interested (Ntourou, Kalogiannakis, & Psycharis, 2021; Frymier et al., 1996). Additionally, effective approaches to impart computer programming frequently rise pupils' interest in programming (Wolz, Pearson, Stone, Switzer, & Pulimood, 2011; Denner, Werner, Campe, & Ortiz, 2014). Students who are interested in programming or who have a good attitude toward teamwork may have more

programming empowerment. Hidi (2006), defines programming interest as a psychological condition in which an individual exhibits positive feelings, concentration, greater focus, or a long-term relatively proclivity to programming actions. A pupil who is more interested in a domain like programming than others sees it as more than just a way to achieving positive outcomes. Instead, students are intrigued by it and investigate it in order to learn more about its implications and effects (Ryan & Deci, 2017, self-determination theory). Furthermore, such a pupil has a greater internal locus of causality than other pupils (Dimmock, Magaraggia, Jackson, & Podlog, 2013).

As a result, students perceive the job more easily (Gattiker, 1992), are eager to spend more time on it (Ryan & Deci, 2017), have higher self-efficacy in programming (Venkatesh & Venkatesh, 2000), and grow greater competency in it (Venkatesh & Speier, 2000).When faced with task difficulties or challenges, such a student is more likely than others to see it as a challenge, persist in finding innovative solutions, and underestimate the drawbacks (Senko & Harackiewicz, 2005; Ryan & Deci, 2017). As a result, computer science and information technology courses offer students the chance to improve the computational thinking skills they will need to function in today's society, which is highly dependent on computation. Information Technology and College Computer Science introductory courses have a high attrition rate and are generally considered to be challenging (Watson & Li, 2014; Robins, Rountree, & Rountree, 2003). In recent years, as interest in making coding available to every college student has increased, a growing number of young individuals are joining college courses with some programming background and programming language familiarity (Lye & Koh, 2014), which could help to increase computer science and information technology persistence rates.

Aside from the growing programming utility and analytical thought in our daily lives, information technology and computer science are likewise a rapidly growing and high-paying sector in which, traditionally, the availability of properly trained employees has failed to keep up with demand. As stated by the Bureau of Labor Statistics (2014), through 2022, eight of the ten jobs in the information technology and computer fields are projected to rise at a rate greater than eleven percent. Despite this rapid growth, it is well known that females continue to be underrepresented in the profession, with the gender gap between men and women entering the field widening. Consistent with the Taulbee study of Computing Association Research from 2013–2014, nearly 86 percent of bachelor's degree recipients in the United States in computer science in 2014 were men. Though computer science bachelor's degrees' reported number increased by fourteen percent in 2014 from 2013, women's proportion earning computer science degrees decreased during that period (Zweben & Bizrot 2015). Recently, the number of students offering computer science and information technology has increased over the last decade (Cuny, 2012). However, in Ghanaian senior high schools, students are rarely exposed to programming and also lack meaningful learning opportunities. As a result, the study sought to examine college students' interest in programming at Kwame Nkrumah University of Science and Technology (KNUST).

Statement of problem

Regardless of age, a novice programmer encounters several hitches while trying to study concepts of programming. For instance, selection structures, variables, recursion, loop structures, pointers, arrays, parameters, references, input/output/handling, structured data types, using language libraries, and error handling (Lahtinen et al. 2005). Computer programming carries young kids into interaction with basic concepts of programming in a suitable developmentally manner, with stress on young users' skills and needs. In Ghanaian universities, students find it difficult to learn computer programming. This has resulted in a loss of interest in learning computer programming. Empirically, several studies have been done on students' interest in programming in developing and developed countries (Ntourou et al., 2021; Papadakis, Kalogiannakis, & Zaranis, 2016; Papadakis, 2016; Kanaki & Kalogiannakis, 2018) but very limited in Africa. Currently, there are limited studies on the effect of age on college students' interest in programming. To bridge the gap in literature, the study sought to investigate college students' interest in programming with age and gender as a factor.

Research questions

The following questions was addressed in the study;

1. What is college students' interest in programming?
2. Does gender has effect on college students' interest in programming?
3. Does age have an effect on college students' interest in programming?

Literature Review

Embedding computational thinking in science courses

The computational nature of current science necessitates the incorporation of computational thinking into science textbooks. As said by Weintrop et al. (2016), incorporating computational tools and techniques into science classes allows students to gain a more realistic understanding of the content of scientific areas, as well as better prepare them for appropriate career pathways. From a pedagogical standpoint, the deliberate application of computational tools and competencies can help students better understand science subject (Wilensky et al. 2014; Sengupta et al. 2013). The converse is also true; science classes give a valuable background and a diverse set of problems to which computational thinking can be applied (Wilensky et al. 2014; Jona et al. 2014). As said by Weintrop et al. (2016), this viewpoint contrasts noticeably from isolating the teaching of computational thinking in a separate

course, where, in most cases, real-world problems and applications are not represented in the assignments students are given. According to Blikstein (2013), the motivation for diversified and purposeful involvement in computational and scientific courses is most likely established if the themes of the assignments are realistic and pertinent to the actual world. As a result, as said by Weintrop et al. (2016), we advocate for an educational method that includes computational programming in science contexts.

Computer Programming

According to Kurkovsky (2007), when CS is compared to other academic subjects, students feel it to be much more difficult. As said by Malan and Leitner (2007), it's considered a "nerdy" and "not a cool course to take". Learning a program is seen as a difficult task. Many new programmers have difficulty learning programming principles and using such concepts in code writing, according to data. As said by Roy et al. (2012), the syntactic subtleties of a language, as well as the computational ideas, are difficult for novice programmers to grasp. Furthermore, according to Guzdial (2004), they have difficulties with basic algorithmic structures like loops, conditions, and the creation of programs using them. Moreover, for decades, enterprise-level languages have been used to introduce programming like C, Java, and C++ in high schools and universities. Programming is a subject that students study at school that aims to equip students with skills and knowledge (Papadakis & Kalogiannakis, 2019). Writing a computer program is a progressive "hands-on activity" for learners to secure an essential skill in their programming career. For instance, students need to gain proficiency with the essential punctuation and then the semantics, structure, and style step by step. During lesson delivery, the instructor will not hold up learners to completely understand programming (Ntourou et al., 2021). Writing computer programs is not just a matter of varying levels of expertise, but additionally, different aptitudes. Writing computer programs is a procedure that starts with interpreting an algorithm into program code. The difficult part is to decipher the required specification into an algorithm. A good algorithm delivers the correct program. Along these lines, students are expected to be talented regarding every procedure: structuring the algorithm, interpreting the algorithm into program language, and composing the programming language with the right punctuation (Kanaki & Kalogiannakis, 2018).

Programming as a course is offered from semester by a semester at the university. Mostly, the first semester for new students, second and third semester for those who have completed the first semester (Papadakis, 2016). Learners who do not perform in the assessment must quit the subject anyway. Also, individuals above the expected mark will continue to the subsequent class. Lamentably, certain students even though they breezed through the assessment, however, they develop no confidence in programming. At the point when they take the next course, they will confront an issue where they have to gain proficiency with another thing just as enhancing their coding ability. The language programming used in indicating a programming introductory course likewise is an issue (Papadakis & Kalogiannakis, 2019).

Programming class is to show the student the most proficient method to program. Students accept that writing computer programs is a challenging thing to learn to have a good grade. But we cannot anticipate that students should be talented in programming just within 14 weeks. Anyway, it can take somebody 10 years to be a gifted software engineer (Rahmat et al., 2012).

College Students' Interest in Programming

An individual's sentiment whose attention, concern, or interest is especially connected by something. The individual has an incredible enthusiasm for the verse of Donne. Something that worries, includes, draws the attention of, or excites the interest of an individual. These inclinations are theory and chess (Rico et al., 2011). A study on pedagogy adapted to create interest in pupils offering a programming course by Venugopal-wairagade (2016). With versatile application improvement getting one of the most looked for after decisions by software engineering students, designers and organizations, a teacher training a seminar on the equivalent to a bunch of fledglings ought to guarantee that the most ideal instructing learning systems are received to create enthusiasm among learners in the study room (Fresko & Alhija, 2015). The creator utilized a mixed methodology comprising of customary, m-learning and e-learning systems. These thusly, comprised of different instructing and assessment techniques (Kamenez, Smirnova, Vaganova, Bystrova, & Tsarapkina, 2019). Through this paper, the creator examines the effect of the strategies embraced as a major aspect of the analysis, on students' enthusiasm and their particular execution. The example comprised of twenty-two postgraduate software engineering pupils. Their reactions to the managed instruments, their learning materials and scores (LMS that includes laptops and projectors) get to logs were broke down.

The creator had the option to accomplish factually huge outcomes concerning the students' advantage and their presentation, for the most part, two methodologies coordination of LMS, and showing the points by building up an application in different stages all through the length of the course. It was discovered that concerning venture work, persistent association with the teacher was an ideal and acknowledged methodology by the pupils. The research additionally uncovered the misinterpretations that a teacher might have while teaching computer programming (Papadakis, 2016). The outcome of this investigation might be utilized by educators showing any computer programming language in advanced education. Toward the finish of the research, it was discovered that pupils, all in all, comprehend following coding rules' significance once composing a program, and it has no association with their programming ability.

It was likewise discovered that once utilizing laptops in class by the student may, to a little degree, rely upon the convenience of the laptop (Kamenez et al., 2019). The investigation likewise uncovered that the students' excitement need not demonstrate that they are enthused about getting their work assessed

consistently (or week after week). In any case, it was seen that the normal reaction to the inquiry in regards to week by week entries and assessments was 1.45, which shows that the majority of the students that shaped the example favour week by week groups instead of presenting the work toward the finish of the semester with no week after week collaboration (Fresko & Alhija, 2015). One reason for this could be that they need to stay away from the very late problem and this causes them to take a shot at their venture normally in a progressively productive way. It was additionally discovered that once coordinating e-learning in the teaching method, if a pupil is effectively occupied with LMS utilization, it might prompt their insight in specific parts of the course, for this situation, creating Android applications (Ntourou et al., 2021).

As the connection between the two factors is frail to direct, we can say that LMS usage might aid in producing the course enthusiasm. A significant perception that was made throughout the research was that there exists a solid positive connection between pupils' marks and their reaction to stage shrewd realizing once instructing programming (Papadakis, Kalogiannakis, & Zaranis, 2016). This demonstrates if the educator encourages a programming course utilizing a couple of uses and builds up the application by circulating the advancement of the highlights all through the semester (Papadakis & Kalogiannakis, 2019). The students might have the option to take a gander at the ideas from a more extensive view and might have the option to relate the ideas with each other since all the ideas are constituents of one entire framework. This might be a superior methodology than showing one idea and afterwards showing another idea, with no connection between the two. The consequences of this investigation might be utilized by educators learning computer programming or any technology building course.

According to Witherspoon *et al.* (2016), students' programming experience is linked to major variations in motivation to seek added opportunities of programming. In other words, pupils who perceived themselves as being extra actively associated with programming were likely more to be motivated to seek added experiences of programming. These findings are in line with those of further active studies learning, which show that teaching that explicitly includes pupils in the process of learning will enhance pupil attitudes toward learning (Casad & Jawaharlal 2012; Alemdar & Rosen 2011; Smith et al. 2014; Prince 2004). It also backs up arguments that cultivating a sense of self as a "programmer" will encourage pupils to seek added opportunities of programming (Worsley & Blikstein 2012; Barton & Tan 2010). The current research discovered that there were no substantial variations in motivation to follow programming between pupils who collaboratively programmed and those who programmed independently. This seems to be at odds with research that suggests that in robotics contexts, teamwork can boost student motivation (Mitnik et al. 2008; Eguchi 2015).

Age and programming interest

Demographics have also been related to empowerment in previous studies. Mature pupils frequently see subjects in school as less important and believe they are less capable than younger students. Older students face more decontextualized teaching than younger pupils, which several find less useful or important in their lives every day. Consequently, they frequently regard subjects in school as less significant (Lepper, Iyengar, & Corpus, 2005). Also, as pupils grow older, they grow into extra mindful of their relative and peers capabilities (Dweck, 2000), motivation achievement, general school-related attitudes, and lowering their domain-specific attitudes (Chiu & Klassen, 2010). Instead of an objective decline in skill or performance, mature pupils' perceived competency might decline because of uncertainties coming from less instruction personalized and improved educational pressures perceptions (Pajares & Cheong, 2003). As a result, mature pupils are likely to regard computer programming as less competent and less meaningful than younger students.

Gender and programming empowerment

Boys are more interested in programming computers than children, owing in part to common media assumptions. Male computer specialists are portrayed in the mass media as both girls' anti-role models and boys' role models, inspiring boys to pursue programming computers while uninspiring girls (Master, Meltzoff, & Cheryan, 2016). Classroom children with computer science stereotypical objects (video games, Star Trek posters) displayed less interest in the subject than classroom children without those objects (Cheryan, Steele, Davies, & Plaut, 2009). As a result, we anticipate that boys will be more interested in programming computers than girls. Boys are interested more than girls are in programming.

Physical Computing and Computational Experiment

Physical computing is said to be useful to people of all ages because it gives a high level of learning, encourages engagement, interest, and sharing of ideas, fosters constructive and creative learning, and cultivates creativity and imagination (Przybylla & Romeike, 2014; Fokides & Papoutsi, 2020). However, the goals to be attained as well as the Physical Computing learning environment must be carefully chosen, since it may be both enjoyable and challenging. It involves extremely taxing tasks that increase cognitive load and may result in a dissatisfying learning experience (Jin, Kearns, & Haynie, 2016). The concept of computational experiment is included in this context, where modeling and computers replace the traditional experimental set-up and simulation replaces the traditional experiment (Zacharia & Michael, 2016; Zacharia, 2007; Psycharis, 2011). It is particularly intriguing because it refers to real issue solving and stimulates students to produce predictions and hypotheses that support the growth of scientific process. It mixes Physics, Math, and the theory of information (Psycharis, 2011).

We may confidently presume that it aids in the development of critical thinking skills and a good attitude toward science (Psycharis et al., 2014; Psycharis, 2013). Many scientists propose using computer experimentation when there is no laboratory available and the experiment is unsafe or needs complex and time-consuming processes (Zacharia, 2007). These types of experiments are significantly easier to employ in the classroom because they need less space and time, do not disturb the classroom's normal operation, and can be repeated much more readily than traditional experiments, giving students more experience (Zacharia, 2007; Klahr et al., 2007). However, Klahr et al. (2007) highlighted that natural materials should be used in regions where the senses are required or where contact with the real world is required. To summarize, the ultimate goal of teaching programming should be to maximize the effectiveness of the learning process by utilizing the potential of both experimental methods; it is therefore advisable to combine computational experiment with classical experiment when teaching programming, as the two are complementary (Zacharia & Michael, 2016; Zacharia, 2007; Jaakkola & Nurmi, 2008).

Methodology

Survey design was used for the study with quantitative methodologies. The study was conducted in Ghana's Ashanti Region at KNUST. The population of the study comprise of Computer Science and Information Technology students at KNUST. The total population was 3,680 college students of Information Technology and Computer Science. In choosing KNUST as the study area, we used purposive sampling technique since there are more than 30 tertiary institutions in Ghana. The sample size was 368 where we used a systematic sampling technique to select participants for the study (Celestine & Nonyelum, 2018). The sample size was calculated as 10% of the population which is 368 participants for the study.

Method of data collection

The study used close ended questionnaire for data collection (Ngitoria, 2014; Kataria, Krishna, Tyagi, & Vashishat, 2019). A Likert scales of five to strongly disagree from strongly agree was applied to measure the items in the questionnaire. Items for the questionnaire were adopted and modified from past studies. Items in the questionnaire to be applied to respond to the research questions were centered on college pupils' interest in programming and the effect of gender and age. Before the use of questionnaire for data collection, a letter of permission and informed consent was sent to the Human Resource Manager and the Head of Department at technology department. Approval form was gotten from the Human Resource Manager and the Head of Department at technology department before the study was conducted.

Validity and Reliability

The validity and reliability of the study were assured. A pilot study was conducted at Presbyterian University College to assess the study's reliability and validity. An application letter and consent form were used to seek permission for the pilot study to be done at Presbyterian University College and use computer science students. After the data collection of the pilot study, we coded and edited it in the Statistical Package for the Social Sciences (SPSS). After that we tested the study's reliability and the Cronbach's Alpha was .890. The pilot study helped to make necessary corrections to the items in the questionnaire.

Data examination

Analysis of the study was done using one statistical tool, called SPSS (Baran, Bilici, Sari, & Tondeur, 2019). To analyse college students' interest in programming, descriptive statistics using skewness, mean, kurtosis, and standard deviation. To analyse college students' interest in programming with regards to gender and age, descriptive statistics like cross tabulation was used. Ethics was considered and observed throughout the study.

Ethical Considerations

The information obtained from respondents were confidential by ensuring that the data collection did not include their names and other information that could show their identity (Petousi & Sifaki, 2020). To find answers to the research questions, respondents were made to understand their positions in the data collection activity. Respondents were given the option of opting out of the exercise in any way that would affect them in order to avoid forcing questionnaires on them. The above-mentioned methods and procedures were used in the search for the appropriate data for the study captured in the next chapter.

Results

The study's purpose was to assess college pupils' interest in programming. The analysis of the results was coded in SPSS, such that college students' interest in programming was coded as CSIIP.

Reliability Test

Table 1 shows that sixteen (16) items with a Cronbach's Alpha of 0.869 (86.9% consistent good result) for gender, age and college students' interest in programming. This means that the variables were reliable.

Table 1: Reliability Statistics

Reliability Statistics

Cronbach's Alpha	N of Items
.869	16

Source: Researcher's fieldwork, (2020)

Gender and college students' interest in programming

One of the specific objectives of the study was to examine college students' interest in programming with gender playing a major role in determining college students' interest in programming. The results revealed that three hundred and fifty-two (352) students representing 96% of the participants were males and sixteen (16) college students representing 4% of the participants were females. Table 2 gives the presentation of the results that satisfy the objectives stated. Table 2 shows that twenty (20) students representing 5% of the males strongly agreed that they have interest in programming. One hundred and seventy-one (171) college students representing 47% of the males agreed that they have interest in programming. Eighty-eight (88) students representing 24% of males slightly agreed that they have interest in programming. Fifteen (15) college students representing 4% of males disagreed that they have interest in programming. Fifty-eight (58) college students representing 16% of males strongly disagreed that they have interest in programming. Also, eight (8) college students representing 2% of females strongly agreed that they have interest in programming and eight (8) college students representing 2% of females strongly disagreed that they have interest in programming.

Table 2: Gender: * CSIIP

		CSIIP					Total
		strongly agree	agree	slightly agree	Disagree	Strongly disagree	
Gender	Male	20	171	88	15	58	352
	Female	8	0	0	0	8	16
Total		30	175	88	15	60	368

CSIIP refers to college students' interest in programming

Source: Researcher's fieldwork, (2020)

Age and college students' interest in programming

The second study's specific objectives was to examine college pupils' interest in programming with age playing a major role in determining college students' interest in programming. Table 3 gives the presentation of results that satisfy the second objective stated. Table 3 shows that all the respondents of the study were between the ages of 22 and 35 years. Furthermore, the results show that twenty-eight (28) college students representing 7% of the participants strongly agreed that college pupils have programming interest. One hundred and seventy-one (171) college students representing 47% of the participants agreed that college pupils have programming interest. Eighty-eight (88) college students representing 24% of the participants slightly agreed that college pupils have programming interest. Fifteen (15) college students representing 4% of the participants disagreed that college pupils have programming interest. Sixty-six (66) college students representing 18% of the participants strongly disagreed that college pupils have programming interest.

Table 3: Age * CSIIP

	CSIIP					Total
	strongly agree	agree	slightly agree	Disagree	Strongly disagree	
Age 22-35 years	28	171	88	15	66	368
Total	28	171	88	15	66	368

CSIIP refers to college students' interest in programming

Source: Researcher's fieldwork, (2020)

Table 4 presents the results of the study. Items used to measure college students' interest in programming were coded as CSIIP. Table 4 demonstrated that "I feel I belong in programming" (CSIIP 6) had the highest mean of 3.0571 which means that, students feel that they belong in programming was the highest interest in programming to students. The average distance a score was from the mean was 1.02795, signifying the dispersion measure (standard deviation) which spread widely the distribution. I feel no tiredness when programming (CSIIP 5) had the second-highest mean of 2.8342 which means that, students feel no tiredness when programming was the second-highest interest in programming to students. The average distance a score was from the mean was 1.13753, signifying the dispersion measure (standard deviation) which spread widely the distribution. Programming is interesting to me (CSIIP 8) had the third-highest mean of 2.8207 which means that programming is interesting to me was the third-highest interest in programming to students. The average distance a score was from the mean was 1.17902, signifying the dispersion measure (standard deviation) which spread widely the distribution. I like programming (CSIIP 1) had the fourth highest mean of 2.7283 which means that

students like programming was the fourth-highest interest in programming to students. The average distance a score was from the mean was 1.19356, signifying the dispersion measure (standard deviation) which spread widely the distribution.

I want to program every day (CSIIP 3) had the fifth-highest mean of 2.4973 which means that, students desire to program every day was the fifth-highest interest in programming to students. The average distance a score was from the mean was 0.92202, signifying the dispersion measure (standard deviation) which spread widely the distribution. I do not get discouraged from setbacks in programming (CSIIP 9) had the sixth-highest mean of 2.4484 which means that students do not get discouraged from setbacks in programming was the sixth-highest interest in programming to students. The average distance a score was from the mean was 0.98077, signifying the dispersion measure (standard deviation) which spread widely the distribution. I become comfortable when programming (CSIIP 4) had the seventh-highest mean of 2.3804 which means that, students become comfortable when programming was the seventh-highest interest in programming to students. The average distance a score was from the mean was 0.81662, signifying the dispersion measure (standard deviation) which spread widely the distribution.

I look forward to taking programming (CSIIP 7) had the eighth-highest mean of 2.0924 which means that students look forward to taking programming was the eighth-highest interest in programming to students. The average distance a score was from the mean was 0.76514, signifying the dispersion measure (standard deviation) which spread widely the distribution. I am happy when programming (CSIIP 2) had the last mean of 2.0163 which means that, students are happy when programming was the last interest in programming to students. The average distance a score was from the mean was 0.75986, signifying the dispersion measure (standard deviation) which spread widely the distribution. The skewness was from 0.050 to 0.828 which means, the variable was normal sufficiently. The kurtosis of six items was less than 0 which means that it had fewer outliers relative to normal distribution. Also, the kurtosis of three items was greater than 0 and less than 3 which means that it had few relatively outliers and scores were clustered more around the mean.

Table 4: College Students' Interest in Programming

	N	Minimum	Maximum	Mean	Std. Deviation	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
CSIIP1	368	1.00	5.00	2.7283	1.19356	.828	.127	-.357	.254
CSIIP2	368	1.00	5.00	2.0163	.75986	.722	.127	1.662	.254
CSIIP3	368	1.00	5.00	2.4973	.92202	.050	.127	-.460	.254
CSIIP4	368	1.00	5.00	2.3804	.81662	.314	.127	-.068	.254
CSIIP5	368	1.00	5.00	2.8342	1.13753	.117	.127	-.834	.254
CSIIP6	368	1.00	5.00	3.0571	1.02795	.385	.127	-.825	.254
CSIIP7	368	1.00	5.00	2.0924	.76514	.576	.127	.694	.254
CSIIP8	368	1.00	5.00	2.8207	1.17902	.573	.127	-.690	.254
CSIIP9	368	1.00	5.00	2.4484	.98077	.642	.127	.059	.254

Source: Researcher's fieldwork, (2020)

Discussion On Findings

College students' interest in programming

The study's results on college students' interest in programming revealed that student feels that they belong in programming was the highest interest in programming to students. The research is comparable to that of Chen et al. (2018) who used the same item and found that it has the third-highest factor loading of 0.80. Students feel no tiredness when programming was the second-highest interest in programming to students. The research is comparable to that of Chen et al. (2018) who found that students do not get disheartened from setbacks which had a factor loading of 0.62. Programming is interesting to me was the third-highest interest in programming to students. The research is comparable to that of Chen et al. (2018) who used the same item and found that it has a higher factor loading of 0.84. Students like programming were the fourth-highest interest in programming to students. The research is comparable to that of Chen et al. (2018) found that students appreciate programming and that they learn with their peers which had a higher factor loading of 0.43. Students' desire to program every day was the fifth-highest interest in programming to students. Students do not get discouraged from setbacks in programming was the sixth-highest interest in programming to students.

The research is comparable to that of Chen et al. (2018) who found that students do not get disheartened from setbacks which had a factor loading of 0.62. Students become comfortable when programming, which is the seventh-highest interest in programming among students. The research is comparable to that of Chen et al. (2018) found that students feel calm with computer science which had a higher factor loading of 0.72. Students look forward to taking programming classes, which was the eighth highest interest in programming among students. The research is comparable to that of Chen et al. (2018) who used the same item and found that the item had the second-highest factor loading of 0.83. Students feel happy when programming is the last interest in programming (Mazaya, 2019). The research is comparable to that of Chen et al. (2018), who found that students enjoy learning it, which had a higher factor loading of 0.73.

Age and Gender of college students interest in programming

The findings revealed that the total number of males who have interest in programming are greater than females. The results of the study is similar to that of Master, Meltzoff, & Cheryan (2016), Doerschuk et al. (2007), Cheryan, Steele, Davies, & Plaut (2009), Modekurty et al. (2014), and Hamner et al. (2008). Also, the study revealed that the total number of males who study programming far outweigh the total number of females. The results is similar to that of Zweben & Bizrot (2015), The College Board (2014), Hendricks et al. (2012), Gal-Ezer & Stephenson (2009), Doerschuk et al. (2007), Witherspoon *et al.* (2016), and Zweben & Bizrot (2015). Furthermore, the study revealed that the total number of college students studying programming are within the ages of 22 and 35 years. The results of the study is similar to the results of Lepper, Corpus, & Iyengar (2005) and Dweck (2000). Moreover, the total number of college students who have interest in programming are within the ages of 22 and 35 years. The study's results is comparable to the results of Chiu & Klassen (2010) and Pajares & Cheong (2003).

In Conclusion

In conclusion, the study shows that the majority of college students have an interest in programming. The study found that the total number of males who have an interest in programming is greater than females, and that the total number of males who study programming far outweighs the total number of females. The study revealed that the total number of college students studying programming are within the ages of 22 and 35 years and that the total number of college students who have interest in programming are within the ages of 22 and 35 years.

Recommendation

It is recommended that females should be encouraged to study programming at the various institutions that provide programming courses to students. It is recommended that future studies should be done on

the factors that influence females' low participation in studying computer programming.

Abbreviations

Kwame Nkrumah University of Science and Technology (KNUST)

Learning materials and scores (LMS)

Statistical Package for the Social Sciences (SPSS)

College students' interest in programming (CSIIP)

Declarations

Availability of data and material: The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Funding: The study was funded by us

Acknowledgement: I give thanks to Beatrice Ampofo for proofreading the work.

Competing interests: The authors declare no competing interests.

References

1. Alemdar, M., Rosen, J. H. Introducing K-12 teachers to LEGO Mindstorm Robotics through a collaborative online professional development course. Paper presented at the Annual, Conference, & Exposition (2011). Vancouver, BC. https://peer.asee.org/18169. Accessed 15 May 2021
2. Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). *A K-6 Computational Thinking Curriculum Framework- Implications for Teacher Knowledge* (19 vol., pp. 47–57). Edu Tech & Soc3
3. Barton, A. C., & Tan, E. (2010). We be burnin'! Agency, identity, and science learning. *The J L Sci*, 19(2), 187–229. doi:10.1080/10508400903530044.
4. Blikstein, P. (2013). Digital fabrication and Bmaking in education: The democratization of invention. In Walter-Herrmann, J., & Büching, C. (Eds.), *FabLabs: Of machines, makers, and inventors* (pp. 1–21). Bielefeld: Transcript Publishers
5. Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. 2012 Annual Meeting of the American Educational Research Association (AERA'12), Canada
6. Bureau of Labor Statistics, U.S. Department of Labor (2014). Occupational outlook handbook, 2014-15: Computer and Information Research Scientists. http://www.bls.gov/ooh/computer-and-information-technology/computer-and-information-research-scientists.htm. Accessed 5 Mar 2021

7. Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *J Med Lit Edu*, 4(2), 121–135
8. Casad, B. J., & Jawaharlal, M. (2012). Learning through guided discovery: an engaging approach to K-12 STEM education. Paper presented at ASEE Annual Conference, San Antonio, Texas. https://peer.asee.org/21643. Accessed 2 Feb 2021
9. Cheryan, S., Plaut, V. C., Davies, P. G., & Steele, C. M. (2009). Ambient belonging: How stereotypical cues impact gender participation in computer science. *J Pers and Soc Psych*, 97(6), 1045–1060
10. Chiu, M. M., & Klassen, R. M. (2010). Relations of mathematics self-concept and its calibration with mathematics achievement. *Learn and Inst*, 20, 2–17
11. Computer Science Teachers Association, Research Committee (CSTA) (2009). CSTA national secondary school computer science survey, 2009. https://csta.acmorg/Research/sub/Projects/ResearchFiles/CSTASurvey09CSResults_DCarter.pdf. Accessed 3 Apr 2021
12. Cuny, J. (2012). Transforming high school computing: a call to action. *ACM Inroads*, 3(2), 32–36
13. Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *J Res Tech in Edu*, 46(3), 277–296
14. Deschryver, M. D., & Yadav, A. (2015). Creative and computational thinking in the context of new literacies: working with teachers to scaffold complex technology mediated approaches to teaching and learning. *J Tech and T Edu*, 23(3), 411–431
15. Dimmock, J., Jackson, B., Podlog, L., & Magaraggia, C. (2013). The effect of variety expectations on interest, enjoyment, and locus of causality in exercise. *Mot and Emo*, 37(1), 146–153
16. Doerschuk, P., Liu, J., & Mann, J. (2007). Pilot summer camps in computing for middle school girls. *ACM SIGCSE Bulletin*, 39(3), 4. doi:10.1145/1269900.1268789.
17. Dweck, C. S. (2000). *Self-theories: Their role in motivation, personality, and development*. Psychology Press
18. Eguchi, A. (2015). RoboCup Junior for promoting STEM education, 21st century skills, and technological advancement through robotics competition. *Rob and Aut Sys*, 75, 692–699. doi:10.1016/j.robot.2015.05.013.
19. Fresko, B., & Alhija, F. N. A. (2015). Induction seminars as professional learning communities for beginning teachers. Asia-Pacific J T Edu, 43(1):36–48. https://doi.org/10.1080/1359866X.2014.928267. Accessed 20 Feb 2021
20. Fokides, E., & Papoutsi, A. (2020). Using Makey-Makey for teaching electricity to primary school students. A pilot study. *Edu and Inf Tech*, 25(2), 1193–1215. doi:10.1007/s10639-019-10013-5
21. Frymier, A. B., Shulman, G., & Houser, M. (1996). The development of a learner empowerment measure. *Comm Edu*, 45, 181–199
22. Gal-Ezer, J., & Stephenson, C. (2009). The current state of computer science in US high schools: a report from two national surveys. Computer Science Teachers Association website,

https://csta.acm.org/Research/sub/Projects/ResearchFiles/StateofCSEDHighSchool.pdf. Accessed 26 Apr 2021

23. Gattiker, U. E. (1992). Computer skills acquisition: A review and future directions for research. *Journal of Management*, 18(3), 547–574
24. Guzdial, M. (2004). In Fincher, S., & Petre, M. (Eds.), *Programming environments for novices*. London: Computer Science Education Research, Taylor & Francis
25. Hamner, E., Lauwers, T., Bernstein, D., Nourbakhsh, I., & DiSalvo, C. (2008). Robot diaries: broadening participation in the computer science pipeline through social technical exploration. Paper presented at the AAAI Symposium on Using AI to Motivate Greater Participation in Computer Science. http://www.aaai.org/Papers/Symposia/Spring/2008/SS-08-08/SS08-08-008.pdf. Accessed 21 June 2021
26. Hendricks, C. C., Alemdar, M., & Ogletree, T. W. (2012). The impact of participation in VEX robotics competition on middle and high school students' interest in pursuing STEM studies and STEM-related careers. Paper presented at the ASEE Annual Conference, San Antonio, Texas. https://peer.asee.org/22069. Accessed 10 June 2021
27. Hidi, S. (2006). Interest: A unique motivational variable. *Edu Res Rev*, 1(2), 69–82
28. Jaakkola, T., & Nurmi, S. (2008). Fostering elementary school students' understanding of simple electricity by combining simulation and laboratory activities. *J Comp Ass Lear*, 24(4), 271–283. doi:10.1111/j.1365-2729.2007.00259.x
29. Jin, K. H., Haynie, K., & Kearns, G. (2016). Teaching Elementary Students Programming in a Physical Computing Classroom [Poster presentation]. 17th Annual Conference on Information Technology Education, Boston, Massachusetts, USA. doi:10.1145/2978192.2978238
30. Jona, K., Wilensky, U., Trouille, L., Horn, M. S., Orton, K., Weintrop, D., & Beheshti, E. (2014). Embedding computational thinking in science, technology, engineering, and math (CT-STEM). In future directions in computer science education summit meeting, Orlando, FL
31. Kamenez, N. V., Smirnova, Z. V., Vaganova, O. I., Bystrova, N. V., & Tsarapkina, J. M. (2019). Development of Instructing Techniques in Professional Training. *Int J Mech Eng and Tech (IJMET)*, 10(2), 899–907
32. Kanaki, K., & Kalogiannakis, M. (2018). Introducing fundamental object-oriented programming concepts in preschool education within the context of physical science courses. *Educ Inf Tech*, 1–26. doi:10.1007/s10639-018-9736-0
33. Klahr, D., Triona, L. M., & Williams, C. (2007). Hands on what? The relative effectiveness of physical versus virtual materials in an engineering design project by middle school children. *J of Res in Sci Tea*, 44(1), 183–203. doi:10.1002/tea.20152
34. Kurkovsky, S. (2007). Making computing attractive for non-majors: a course design. *J. Comp. Sci. Coll*, 22(3), 90–97
35. Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. In Acm Sigcse Bulletin 37(3):14-18. New York: ACM

36. Lepper, M. R., Corpus, J. H., & Iyengar, S. S. (2005). Intrinsic and extrinsic motivational orientations in the classroom: Age differences and academic correlates. *J of Edu Psych*, 97(2), 184–196
37. Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K–12? *Comp in H Beh*, 41, 51–61
38. Malan, D. J., & Leitner, H. H. (2007). Scratch for budding computer scientists. in ACM SIGCSE Bulletin, 39(1):223–227, ACM, New York
39. Master, A., Cheryan, S., & Meltzoff, A. N. (2016). Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science. *J of Edu Psych*, 108(3), 424
40. Mitnik, R., Nussbaum, M., & Soto, A. (2008). An autonomous educational mobile robot mediator. *Aut Rob*, 25, 367–382. doi:10.1007/s10514-008-9101-z.
41. Modekurty, S., Fong, J., & Cheng, H. H. (2014). C-STEM girls computing and robotics leadership camp. Paper presented at 2014 ASEE Annual Conference, Indianapolis, Indiana. https://peer.asee.org/20141. Accessed 20 May 2021
42. Ntourou, V., Kalogiannakis, M., & Psycharis, S. (2021). A Study of the Impact of Arduino and Visual Programming In Self-Efficacy, Motivation, Computational Thinking and 5th Grade Students ' Perceptions on Electricity. EURASIA J of Math, Sci and Tech Edu, 17(5):1–11. Retrieved from https://doi.org/10.29333/ejmste/10842
43. Pajares, F., & Cheong, Y. F. (2003). Achievement goal orientations in writing. *Int J of Edu Res*, 39, 437–455
44. Papadakis, S. (2016). Using Scratch and App Inventor for teaching introductory programming in secondary education. A case study Stamatios Papadakis * Michail Kalogiannakis and Nicholas Zaranis Vasileios Orfanakis. *Int. J. Tech Enh Learn*, 8(3/4), 217–233
45. Papadakis, S., & Kalogiannakis, M. (2019). Evaluating a course for teaching introductory programming with Scratch to pre-service kindergarten teachers. *Int J of Tech Enh Learn*, 11(3), doi:10.1504/IJTEL.2019.100478
46. Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: a case study Stamatios Papadakis *, Michail Kalogiannakis and Nicholas Zaranis. *Int. J. Mob Learn and Org*, 10(3), 187–202
47. Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books
48. Prince, M. (2004). Does active learning work? A review of the research. *J of Eng Edu*, 93, 223–231. doi:10.1002/j.2168-9830.2004.tb00809.x.
49. Petousi, V., & Sifaki, E. (2020). Contextualizing harm in the framework of research misconduct. Findings from discourse analysis of scientific publications. *Int J of Sust Dev*, 23(3/4), 149–174. doi:10.1504/IJSD.2020.10037655
50. Przybylla, M., & Romeike, R. (2014). Physical computing in computer science education [Paper presentation]. 9th Workshop in Primary and Secondary Computing Education. https://doi.org/10.1145/2670757.2670782. Accessed 4 Mar 2021

51. Psycharis, S. (2011). The computational experiment and its effects on approach to learning and beliefs on physics. *Comp & Edu*, 56(3), 547–555. doi:10.1016/j.compedu.2010.09.011
52. Psycharis, S. (2013). Examining the effect of the computational models on learning performance, scientific reasoning, epistemic beliefs and argumentation: An implication for the STEM agenda. *Comp & Edu*, 68, 253–265. doi:10.1016/j.compedu.2013.05.015
53. Psycharis, S., Botsari, E., Mantas, P., & Loukeris, D. (2014). The impact of the computational inquiry based experiment on metacognitive experiences, modelling indicators and learning performance. *Comp & Edu*, 72, 90–99. doi:10.1016/j.compedu.2013.10.001
54. Rico, M., Martínez-muñoz, G., Alaman, X., Camacho, D., & Pulido, E. (2011). A Programming Experience of High School Students in a Virtual World A Programming Experience of High School Students in a Virtual World Platform *. Int J of Eng Edu, 27(May 2014):1–9. https://www.researchgate.net/publication/233759773%0D. Accessed 13 July 2021
55. Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Comp Sci Edu*, 13(2), 137–172
56. Roy, K., Rousse, W. C., & DeMeritt, D. B. (2012). Comparing the mobile novice programming environments: app inventor for Android vs. GameSalad. in 2012 IEEE Frontiers in Education Conference (FIE) (FIE '12), IEEE Press, New York, pp.1–6
57. Ryan, R. M., & Deci, E. L. (2017). *Self-determination theory*. New York, NY: The Guilford Press
58. Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Edu and Inf Tech*, 18(2), 351–380
59. Senko, C., & Harackiewicz, J. M. (2005). Achievement goals, task performance, and interest: Why perceived goal difficulty matters. *Pers and Soc Psych Bulletin*, 31(12), 1739–1753
60. Smith, N., Sutcliffe, C., & Sandvik, L. (2014). Code club. Proceedings of the 45th ACM Technical Symposium on Computer Science Education - SIGCSE'14 (pp. 517–522). doi:10.1145/2538862.2538919
61. The College Board (2014). The 10th annual AP report to the nation: subject supplement. http://apreport.collegeboard.org
62. Venkatesh, V. (2000). Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model. *Inf Sys Res*, 11(4), 342–365
63. Venkatesh, V., & Speier, C. (2000). Creating an effective training environment for enhancing telework. *Inter J of H-Comp Stu*, 52(6), 991–1005
64. Venugopal-wairagade, G. (2016). Study of a pedagogy adopted to generate interest in students taking a programming course. 2016 International Conference on Learning and Teaching in Computing and Engineering, 141–146. https://doi.org/10.1109/LaTiCE.2016.27. Accessed 1 July 2021
65. Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice.Edu and Inf Tech,1–14

66. Watson, C., & Li, F. W. B. (2014). Failure rates in introductory programming revisited. Proceedings of the 2014 conference on Innovation & technology in computer science education, Uppsala, Sweden
67. Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *J of Sci Edu and Tech*, 25(1), 127–147
68. Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering computational literacy in science classrooms. *Comm of the ACM*, 57(8), 24–28
69. Wolz, U., Stone, M., Pearson, K., Pulimood, S. M., & Switzer, M. (2011). Computational thinking and expository writing in the middle school. *ACM Trans on Comp Edu*, 11(2), 1–22
70. Worsley, M., & Blikstein, P. (2012). A framework for characterizing changes in student identity during constructionist learning activities. Paper presented at Constructionism 2012, Athens, Greece. https://tltl.stanford.edu/sites/default/files/files/documents/publications/20 1 2.ConstructionismWB.Framework.pdf. Accessed 18 May 2021
71. Zacharia, Z. C. (2007). Comparing and combining real and virtual experimentation: an effort to enhance students' conceptual understanding of electric circuits. *J of Comp Ass Learning*, 23(2), 120–132. doi:10.1111/j.1365-2729.2006.00215.x
72. Zacharia, Z., & Michael, M. (2016). Using Physical and Virtual Manipulatives to Improve Primary School Students' Understanding of Concepts of Electric Circuits. In Riopel, M., & Smyrnaiou, Z. (Eds.), *New Developments in Science and Technology Education* (pp. 125–140). Springer
73. Zweben, S., & Bizrot, B. (2015). 2014 Taulbee survey. Retrieved from the Computing Research Association website, http://cra.org/wp-content/uploads/2015/06/2014-Taulbee-Survey.pdf. Accessed 15 June 2021