Identifying major challenges faced by teachers while teaching computer programming in universities

Md. Rashedul Huq Shamim*

Islamic University of Technology (IUT),
Dhaka, Bangladesh
Email: mrhuq09@iut-dhaka.edu
*Corresponding author

Atebeni Cherotwo

Islamic University in Uganda (IUIU),
Mbale, Uganda
Email: atebenicherotwo@iut-dhaka.edu

Md. Abu Raihan

Islamic University of Technology (IUT),
Dhaka, Bangladesh
Email: maraihan@iut-dhaka.edu

Abstract: In today's information age, learning computer programming is very important for teachers because it provides them with many teaching opportunities. However, teachers face major challenges while teaching computer programming in universities. Identifying these challenges may assist educators in overcoming them; therefore, this study intended to discover the major challenges that teachers face. A purposive sampling method was used in this quantitative study, and 114 teachers responded to the distributed questionnaires. Weighted average and chi-square tests were conducted whereas the results indicated a positive response from the teachers in that they desired to learn but were faced with many challenges in teaching computer programming. Therefore, it's recommended that ongoing pre-service and in-service training accommodate the instructors to build another model for instructional methods, advances, and apparatuses for learning to improve computer programming education. Moreover, it creates the incentive for educator coaches and strategy producers to comprehend computer programming courses' limitations and cost-effectiveness.

Keywords: challenges; computer programming; teaching and communication.

Reference to this paper should be made as follows: Shamim, M.R.H., Cherotwo, A. and Raihan, M.A. (2021) 'Identifying major challenges faced by teachers while teaching computer programming in universities', *Int. J. Smart Technology and Learning*, Vol. 2, No. 4, pp.287–303.

Biographical notes: Md. Rashedul Huq Shamim is an associate Professor of the Technical and Vocational Education Department at Islamic University of Technology (IUT). His research interest includes computer-assisted instruction, mobile learning, ICT integration in education, and STEM education.

Atebeni Cherotwo is a lecturer at Islamic University in Uganda. His research interest includes ICT in education, E-learning, Programming language & integration of ICT in education.

Md. Abu Raihan is a Professor & Head, of the Technical and Vocational Education Department at Islamic University of Technology (IUT). His research interest includes Web-based e-learning, computer-assisted instruction, mobile learning, technology integration in education, and STEM education.

1 Introduction

Computer Programming (CP) is a fundamental course in education in today's digital world. Teachers and learners are thus required to communicate effectively in CP courses. CP is taught to students all over many Universities today (Alghamdi, 2017). Teaching Computer Programming is challenging because it requires teachers to keep a braze with tools, updated pedagogy skills, and communicate the knowledge to students effectively and efficiently. Moreover, students must possess passion and interest to learn Computer Programming (Oroma et al., 2012). CP programs consist of courses spread along with domains such as mathematics, computing, ethics, design, and engineering, etc. Each course complements the other in terms of knowledge set and structure (Hooshyar et al., 2015b). CP will be influenced by the way institutions, universities, and TVET sector operate in reaching out and providing service to their learners, which is one of the major contests being handled by the teachers during teaching computer programming courses (Koryuhina and Shamshina, 2018). This research gives insight in identifying the major challenges faced by teachers during teaching computer programming courses and several innovative approaches that leads to identifying the knowledge gap or skills gap and improving the educating and learning of the PC programming courses where it will point out major challenges, solutions, and recommendations (Tshabalala and Ncube, 2014). CP is a generic term referring to technologies that are being used for collecting, storing, editing, and passing information in various practices, whereas universities providing facilities for computer programming using tools to communicate, and teach (Ghavifekr et al., 2006; Koryuhina and Shamshina, 2016). However, studies indicate that teaching using improved pedagogical methods has a great impact on knowledge and understanding (Oroma et al., 2012). CP plays various roles to cope with major challenges that teachers face during teaching computer programming in universities (Pelgrum, 2001); in the authors' opinion, through a well-organised determination by teachers to identify major challenges and inspire teachers to share ideas to overcome challenges (Tshabalala and Ncube, 2014). Basically, teachers of computer programming courses in universities will be able to address the weakness like; improve teaching skills, select appropriate teaching methods; the outcomes due to effectiveness will result in; the teacher will be able to produce competent students in the area of computer programing courses, improve students learning skills, knowledge gap is eliminated through having competent teachers who are well acquainted with the knowledge of computer programming. Several studies have been reviewed, the literature on identifying major

challenges faced by teachers during teaching computer programming and it has great potentials to enhance teacher's achievements and student learning (Pelgrum, 2001). According to the previous research studies and theories state that the use of computers can help teachers and learners develop more educated hence reduce the workload of direct teaching and give extra time for teachers to utilise in a well-planned way through reaching onto the learners with particular needs (Rahman et al., 2015). Besides that, the teacher also enhances the pedagogical practice that equips them with knowledge, skills, and experience which enables teachers to teach in various platforms of computer programming and able to examine, interpret process and spread content or information to learners without any challenges faced (Tshabalala and Ncube, 2014). The study will be of great value to universities basically in building capacity by eliminating major challenges that deny the progress of good instruction and learning of computer programming courses. Teachers and learners have direct access to all software to support the curriculum and encourage implementation of its use in CP classes in promoting a better teaching-learning process without any major challenges faced (Rahman et al., 2015). Again for quality of education through motivating the teachers together with learners and engagement, enabling, achievement of basic skills, enhancing teacher training and promote students centered learning environment (Oroma et al., 2012). The study, however, will be of great value to policymakers within universities. The findings of this will promote better and effective teaching and learning practices among university teachers (Koryuhina and Shamshina, 2018). The researcher hopes that the results of the study will be of great value for future researchers with interest in examining how to improve on major challenges faced by teachers during teaching computer programming to enable or make teaching-learning effective in universities that will lead to the generation of good ideas for better implementation. The following research question guided the study to gather data that is in line with its objectives.

- What are the major challenges faced by teachers during teaching computer programming? and also what can be done to improve this situation?

2 Background of the study

With an ever-increasing need for talented experts in the 21st century, especially in the computing discipline, educational institutions have been forced to improve on the quality of their teaching practices (Petri and Von Wangenheim, 2017). Computer programming includes explicit specialised parts of programming dialects and imaginative viewpoints to track down the best answers for various issue areas while it envelops a gathering of various instructor understudy procedures that are incorporated to tackle genuine issues; in this learning interaction, we should incorporate great programming improvement practices of examination and plan so the beginner developer teaches himself into creating quality programming (Martínez et al., 2014). A visual programming environment and a game-development approach help novices, middle school students to understand concepts that are problematic in various settings, namely variables, expressions, logic, and loops whereas a game development-based approach is proposed for assessing students' programming knowledge and motivation (Papadakis, 2020). The main aim is to

identify major challenges faced by teachers during teaching computer programming courses in universities, as well as different authors used different methodologies in addressing these major challenges. However, it always emphasises the learning of the basic concepts which focus on constructing further developed abilities. There is a developing agreement that computational reasoning is a principal ability that everybody needs to prevail in our mind-boggling and technological culture. Writing computer programs is an astounding method to create computational reasoning abilities (Papadakis and Kalogiannakis, 2017). Code with project points was to embrace the learning of the essential ideas and constructions by making dynamic visual learning materials for students (Teague et al., 2016). Different investigations have shown that gamification under suitable conditions may establish a climate helpful for learning and lead to huge expansions to understudies' greatest advantage in programming (Papadakis and Kalogiannakis, 2017). Programming language is basic in science and innovation training, it very well may be hard for certain understudies, particularly fledglings. One potential explanation may be the way that programming language, is too perplexing psychological action and dynamic for these understudies to comprehend (Papadakis, 2020). Instructors and scientists have 'battled' with the educating and learning of programming trying to decide approaches and instructional systems to be the most proficient and successful (Papadakis and Orfanakis, 2018). Different investigations have shown that pair programming under suitable conditions may establish a climate helpful for picking up, prompting an expansion of understudy revenue in programming. The outcomes showed that the pair writing computer programs is more proficient than the performance programming, both on working with and supporting understudies' learning and comprehension of essential programming ideas, just as on improving understudies' perspectives toward programming (Papadakis, 2018). Practice is vital in procuring programming abilities; there ought to be space to commit errors and to gain from them. Educators could help their students as they practice, however, programmed evaluation can assist with giving input in various circumstances because of the constraints of human resources through the utilisation of programmed peer appraisal to give understudies freedoms to effectively gain from each other (Carbonaro, 2019). Computer programming is one of the sophisticated courses to teach and learn, it is even more difficult in developing countries that are faced with additional factors in their academics. Thus, the study has a great impact on learning communities in terms of improving problem-solving abilities, enhance self-efficacy, adjust the appropriate pedagogical methods, and decide study methods and learning styles of the learners.

2.1 Challenges in learning computer programming

Programming learning is commonly measured to be difficult, and in which often has high dropout rates (Hooshyar et al., 2015b). However, numerous instructive examination has been completed to distinguish the attributes of fledgling software engineers and to consider the learning interaction and its effects on the various parts of programming (Teague and Lister, 2014). The most favoured technique in programming among beginner understudies is that of 'bricolage': Understudies create programs in the PC, and prone to avoid the periods of investigation and plan (Teague et al., 2016). In the connection with this training, beginner developers think of mental or psychological

issues and wrong thoughts regarding figuring that make it hard to comprehend the working of projects or the formation of calculations. Errors can be perceived to the way that understudies decipher PC programming as correspondence among people (Pelgrum, 2001). According to Ghavifekr et al. (2016), misunderstandings were difficult to modification unless teachers attain a model of computer. For solving major challenges, teachers apply to learn speculations to personal computer (PC) programming, specifically the constructivist learning theory than they said that beginner understudies need to assemble a substantial model of a PC to reduce the troubles of picking up programming (Teague et al., 2016). Furthermore, the expertise in programming requires the acquisition of higher-order thinking abilities, like examination, plan, analogical reasoning, reuse, assessment, and proliferation (Teague and Lister, 2014). Finally, inclusions with mixed learning models, which mix up close and personal learning and Online frameworks or comparative programming, are step by step turning into an appealing alternative as new inventive innovations become logically accessible (Angeli and Valanides, 2009).

Subsequently, regardless of whether some advancement has been made in tackling some learning issues utilising learning hypothesis and data advances, the issues, and inconveniences related to the learning PC programming stay to be explored. A few instructors accept that taking care of the issues related to the learning of programming requires a significant change from old-style instructional techniques to constructivist learning conditions and arranged learning (Alghamdi, 2017), commonly because computer programming is measured as a skill that teachers prerequisite to acquire through an active building process. Furthermore, according to Gomes and Mendes (2007), writing computer programs is in a general sense of social movement as great projects are created not in detachment; all things being equal, they include a connection with others. Programming aptitudes and strategies are found out from a wide assortment of sources; a considerable lot of them are not homeroom-based (Rahman et al., 2015).

2.2 Characteristics of novice programmers

The amateur developers allude to one who needs information and abilities of programming, though various isolating variables have been concentrated in the writing and were additionally reexamined (McCracken et al., 2001). The learners have all the earmarks of being with fractional surface information on projects and for the most part approach programming ‘line by line’ rather than at the level of higher program structures (Agustin et al., 2018). McCracken et al. (2001) reported in their review the troubles of C++ programming by directing an electronic poll for instructors. However, the obvious outcomes show that students appraised having not many challenges than assembled from educators’ answers. This progression is the experimental perceptions of numerous educators; programming learners regularly neglect to perceive their own deficiencies (Amin and Rahimi, 2018). Therefore, no writing found on huge contrasts in discovering that is brought about by classifications like ethnicity or sex yet broad insight and science or numerical capacities seem, by all accounts, to be related to progress at figuring out how to program (Tshabalala and Ncube, 2014). Through the programming course, various teachers behavior’s in challenging a problematic situation can be recognised according to where they mentioned two main types: movers and stoppers (Tshabalala and Ncube, 2014). With everything taken into account, there are viable and ineffectual

tenderfoots, for example, instructors who learn without unreasonable exertion and the individuals who don't learn without unnecessary individual consideration (Teague and Lister, 2014). Naturally, students' very own learning techniques and inspiration influence their accomplishment in picking up programming systems (Hooshyar et al., 2015a).

2.3 Various parts of programming

Picking up programming contains a few exercises, for example, learning highlights, program plan, and program appreciation. With an average methodology in reading material and programming courses, was begun with definitive information about a specific language (Teague and Lister, 2014). However, studies show that it is critical to carry likewise different perspectives to the main programming courses. A few regular deficiencies in amateurs' comprehension of explicit programming language development are introduced (Alghamdi, 2017). The primary cause of trouble doesn't appear to be the sentence structure or comprehension of ideas, yet rather an essential program arranging (Teague et al., 2016). To assist educators with learning both idea information and approaches for their utilisation, built up a critical thinking strategy for a programming course (Vihavainen et al., 2011). In their methodology, language highlights were acquainted with understudies just concerning explicit issues, gradually. This appeared to impact understudies' advancement results and their programming sureness. Valuable conversation on issue-based learning in programming courses can be set up (Čisar et al., 2014). This requires a capacity to follow code to construct a psychological model of the program and anticipate its conduct (Teague and Lister, 2014). This is one of the capacities that could be made by underlining program comprehension and exploring strategies in the programming courses.

2.4 Object-oriented vs. procedural approaches

Article arranged methodology advantages to be the conventional method of abstracting certifiable difficulties. In any case, examines don't appear to help that (Nawrocki et al., 2002). The class design of the research articles situated the program made it a cycle more obvious program substances, however particularly program stream and information stream issues were more clear from a procedural program. Kiczales et al. (1997) recognised some idea challenges relating to the particularly object-arranged worldview. The two students and educators thought about constructors, virtual capacities, and capacity abrogating in the legacy among the most troublesome issues in the article arranged programming (Nawrocki et al., 2002). Interestingly, students considered virtual to be pretty much as simply calmly troublesome rather than educators who considered them as the second serious issue among the ideas recorded in the review. In like, two or three respondents that were showing thing orchestrated language conveyed that they recognise article arranged dialects ought to just be progressed once the understudy has an intensive structure up in procedural programming (Andreoli et al., 1997). In any case, likewise, they saw that educators had more issues with more significant level item arranged territories than sentence structure and style (Van den Akker et al., 2004).

3 Methodology

The researcher was engaged in the work and this study adopted a quantitative research method (Suryani, 2008). Specifically, a descriptive survey method was used to identify major challenges faced by teachers during teaching computer programming courses in universities and the solutions to overcome. A questionnaire is considered the most appropriate tool as recommended (Amin and Rahimi, 2018). In this pursuit, a questionnaire was used to collect data since it allows the respondents to easily express their perception from their own lens. In this research study, purposive sampling was utilised to choose 114 teachers from twelve (12) Universities in Uganda who are from computer science-related domains. However, the sample comprised of both male and female teachers, who participated in providing their views about the findings.

3.1 Demographic information

Table 1 represents the demographic information of the teachers male and female. Out of the total of one hundred and fourteen (114) respondents, seventy-one (71) were males with 62.3% and forty-three (43) were females with 37.7% from the CSE teaching area.

Table 1 Demographic information

	Frequency	*%*
Female	43	37.7
Male	71	62.3
Total	114	100.0

However, out of 114 samples from the 12 universities, a total of 114 teachers responded. The survey was divided into three parts, like as the closed-ended questions; the first one involved questions on the demographic data, background characteristics, the second part included the major challenges facing students while learning computer programming courses where five-point Likert scale was used and the third part was a checklist format questions used to get the related information from the participants. The questionnaire was designed using Google forms to collect the required data through a shared link between the researcher and the target sample. Being an online form, all responses were received in real-time upon the completion of the form by the respondents (Van Den Beemt and Diepstraten, 2016). The research assistants printed and administered some copies of the questionnaire physically to the offline teachers. After collecting the data, the research assistants sent back the scanned copies of the administered questionnaire through email and WhatsApp. The resulted data collected, was compiled and analysed using software known as Statistical Package for Social Sciences (SPSS). Several studies have used online questionnaires to collect data that have yielded good results (Gauch and Moran, 2019). Following the quantitative analysis method, data was collected and analysed from the respondents using the questionnaire and tabulated in the form of the frequencies, percentages, and also each table tabulated, follows its detailed interpretation were the open-ended questions were tabulated using frequency, and percentages. However, Nonparametric Chi-square (χ^2) test and the weighted normal were utilised to discover the centrality of the distinctions of information got and also the questionnaires were analysed by weighted average (WA) meanwhile SPSS version 25 software was used to do a quantitative analysis using weighted average and chi-square (χ^2) test for analysing data

collected from the structured questionnaire. To investigate regardless of whether the assessments of the respondents were measurably critical or not, Chi-square (χ^2) test was used, the significant value was compared with the critical value at 0.05 level of significance, and the degree of freedom was calculated as well but the criteria for five-point Likert scale was interpreted in Table 2.

Table 2 Interpretation of weighted average based on five-point Likert scale

Weighted Average	*Weighted Average Interpretation*
Weight Average ≥ 4.5	Strongly Agree –SA (5)
4.5 >W.A ≥3.5	Agree – A (4)
3.5 > W.A ≥2.5	Undecided – U (3)
2.5 >W.A ≥1.5	Disagree – D (2)
1.5 >W.A ≥0	Strongly Disagree – SD (1)

Table 2 shows the interpretation of weighted average whereby above Weight Average ≥ 4.5 means that the respondents strongly agree and their opinions have high confidence, 4.5 >WA ≥ 3.5 means between 4.5 and 3.5 which indicates that respondents agree, and 3.5 >WA ≥2.5 indicates respondents opinions are undecided and this is where further analysis using chi-square test was important, 2.5 >WA ≥ 1.5 indicates that the respondents disagree with the statement and 1.5 >WA ≥ 0 reveals that they strongly disagree.

4 Results and discussion

The data collected from the respondents were tabulated in the form of frequencies and percentages. The tables were separated and used to explain some of the opinions based on the feedback received from the teachers about the questions. Moreover, Chi-square (χ^2) test and weighted average (WA) were calculated and analysed based on the data collected in the form of a five-point Likert scale and checklist using SPSS version 25.

The researcher characterised the constraints into three levels depending on the weighted average; the confidence included (i) top-level, (ii) mid-level and (iii) lower level which was analysed below.

(i) In the top-level challenges, the researcher identified that the internet which was important to be used besides the lecture notes and textbooks to learn Computer programming, practice Computer Programming by using a personal computer, classroom management (teacher-student relationship, etc.) was successful and supplied necessary resources/course materials that were adequate and updated.

The findings revealed that the internet was being used besides the lecture notes and textbooks to learn effectively computer programming which was identified as the major challenges faced by teachers during teaching computer programming in universities though the facilities of internet use was not available in all of the institutions. The respondents, 54.4% strongly agreed, 38.6% agreed, 2.6% undecided, 2.6% disagreed and 1.8% strongly disagreed. The weighted average of 4.41(4.5 >WA ≥ 3.5) indicated high confidence of the internet was used besides the lecture notes and textbooks to learn computer programming.

Table 3 Challenges face by teachers while teaching computer programming

S/N	*Statement*		*5(SA)*	*4(A)*	*3(U)*	*2(D)*	*1(SD)*	*WA*	χ^2	*Sig. Val*	*Df*
1	Computer Programming courses are difficult to learn	f	9	17	10	37	41	2.26	40.386	.000	4
		%	(7.9)	(14.9)	(8.8)	(32.5)	(36.0)				
2	Practice Computer Programming by using a personal computer.	f	55	48	5	5	1	4.32	121.965	.000	4
		%	(48.2)	(42.1)	(4.4)	(4.4)	(.9)				
3	Unfriendly approaches of my colleagues make me upset	f	11	33	14	43	12	2.89	37.398	.000	4
		%	(9.6)	(28.9)	(12.3)	(37.7)	(10.5)				
4	Internet is used besides the lecture notes and textbooks to learn Computer programming	f	62	44	3	3	2	4.41	140.474	.000	4
		%	(54.4)	(38.6)	(2.6)	(2.6)	(1.8)				
5	Difficulty in managing unruly classes with discipline problems	f	8	48	22	28	7	3.19	50.053	.000	4
		%	(7.0)	(42.1)	(19.3)	(24.6)	(6.1)				
6	Difficulty in choosing appropriate teaching methods	f	5	43	21	35	8	3.02	48.893	.000	4
		%	(4.4)	(37.7)	(18.4)	(30.7)	(7.0)				
7	Classroom management (teacher-student relationship etc.) was successful	f	12	64	28	7	3	3.66	108.895	.003	4
		%	(10.5)	(56.1)	(24.6)	(6.1)	(2.6)				
8	Supplied necessary resources/course materials that were adequate and updated	f	9	65	27	11	2	3.60	112.316	.000	4
		%	(7.9)	(57.0)	(23.7)	(9.6)	(1.8)				

The significant value 0.000 level which was not exactly the Alpha of 0.05 (P<.05), means the null hypothesis, reactions on this assertion was not genuinely critical, was dismissed and the assessments of educators concerning the web were utilised other than the take notes and course books to learn writing computer programs were emphatically concurred on which was measurably significant. Practice programming by utilising a PC was distinguished as the significant difficulties looked at by instructors during encouraging programming in colleges. The respondents, 48.2% strongly agreed, 42.1% agreed, 4.4% undecided, 4.4% disagreed and .9% strongly disagreed. The weighted normal 4.32 (4.5 >WA ≥ 3.5) showed high certainty of training Computer Programming by utilising PCs. The tremendous characteristics 0.000 level which was not actually the Alpha of 0.05 (P<0.05), suggests invalid theory, responses on this affirmation were not truly basic and excused while the evaluations of teachers as for practice Computer Programming by using a PC unequivocally agreed which was genuinely critical. Classroom management (teacher-student relationship etc.) was successfully identified as the major challenges faced by teachers during teaching computer programing in universities in which 10.5% strongly agreed, 56.1% agreed, 24.6% undecided, 6.1% disagreed and 2.6% strongly disagreed. The weighted average of 3.66 (4.5 >WA≥ 3.5) indicated high confidence in classroom management (teacher-student relationship etc.) were successful. The significant values 0.003 level which was less than the Alpha of 0.05 (P<0.05) means the null hypothesis, responses on this statement were not statistically significant, was prohibited and the opinions of teachers regarding classroom management was successful agreed which was statistically significant. Supplied necessary resources/course materials that were adequate and updated and were identified as the major challenges in which 7.9% strongly agreed, 57.0% agreed, whereas 23.7% undecided, 9.6% disagreed, and 1.8% strongly disagreed. The weighted average of 3.60 (4.5 >WA≥ 3.5) indicated high confidence of supplied necessary resources/course materials that were adequate and updated.

In the mid-level challenges, difficulty in managing unruly classes with discipline problems, difficulty in choosing appropriate teaching methods.

The result revealed the difficulty in choosing appropriate teaching methods was identified as the major challenges faced by teachers whereas 7.0% strongly agreed, 42.1% agreed, 19.3% undecided, 24.6% disagreed and 6.1% strongly disagreed. The weighted average of 3.19 (3.5 >WA≥ 2.5) indicated high confidence of difficulty in choosing appropriate teaching methods. The significant values 0.000 level which was less than the Alpha of 0.05 (P<0.05) implies the null hypothesis was rejected and the opinions of teachers regarding trouble in overseeing uncontrollable classes with discipline issues were agreed which was statistically significant. The result revealed that choosing appropriate teaching methods was identified as the major challenges faced by teachers during teaching computer programing in which 4.4% strongly agreed, 37.7% agreed, 18.4% undecided, 30.7% disagreed and 7.0% strongly disagreed. The weighted average of 3.02 (3.5 >WA≥ 2.5) indicated high confidence of difficulty in choosing appropriate teaching methods.

(ii) In the lower-level challenges, the Computer Programming course was difficult to learn, unfriendly approaches of my colleagues make me upset

Table 4 Instructional approaches to improve teaching computer programming courses

S/N	*Statement*		*5(SA)*	*4(A)*	*3(U)*	*2(D)*	*1(SD)*	*WA*	χ^2	*Sig. Val*	*Df*
1	Computer Programming teacher is always available for questions after classes if students have not understood the lecture clearly	f	11	58	17	25	2	3.45	81.823	.000	4
		%	(9.6)	(50.9)	(14.9)	(21.9)	(1.8)				
2	Computer Programming teacher always shows a keen interest in motivating students to accomplish Computer Programming practical's	f	15	58	26	14	1	3.63	81.702	.000	4
		%	(13.2)	(50.9)	(22.8)	(12.1)	(.9)				
3	Computer Programming teacher does not explain beyond the examples in the textbooks he/she is as using	f	12	59	31	8	3	3.61	93.150	.000	4
		%	(10.5)	(51.8)	(27.2)	(7.0)	(2.6)				
4	While teaching a Computer Programming course, the teacher cites local examples to explain the concept to students	f	14	57	34	9	0	3.67	50.281	.000	3
		%	(12.3)	(50.0)	(29.8)	(7.9)	(0)				
5	Teacher gives Computer Programming projects during the course time	f	18	55	31	10	0	3.71	40.737	.000	3
		%	(15.8)	(48.2)	(27.2)	(8.8)	(0)				
6	Teacher complete program successfully in class.	f	14	71	17	11	1	3.75	133.719	.000	4
		%	(12.3)	(62.3)	(14.9)	(9.6)	(.9)				
7	Provide assignments and homework after every Computer Programming lesson	f	20	55	25	8	6	3.66	68.018	.000	4
		%	(17.5)	(48.2)	(21.9)	(7.0)	(5.3)				
8	Teachers feel insufficient in testing and evaluation	f	23	47	23	15	6	3.58	40.737	.000	4
		%	(20.2)	(41.2)	(20.2)	(13.2)	(5.3)				

The result revealed that Computer Programming courses were difficult to learn was identified as the major challenges faced by teachers during teaching computer programming in universities. 7.9% strongly agreed, 14.9% agreed, 8.8% undecided, 32.5% disagreed and 36.0% strongly disagreed. The weighted average of 2.26 (2.5 >WA≥ 1.5) indicated high confidence of Computer Programming courses were difficult to learn.

The result revealed that hostile approaches of my colleagues' make me upset was identified as the major challenges faced by teachers during teaching computer programming in universities. 9.6% strongly agreed, 28.9% agreed, 12.3% undecided, 37.7% disagreed and 10.5% strongly disagreed.

4.1 What can be done to improve this situation?

According to the researcher categorised the constraints into three levels depending on the weighted average confidence which includes (i) top-level, (ii) mid-level, and (iii) lower level which was analysed below.

(i) In the top-level challenges, the teacher did programming successfully in class, teacher gave the Computer Programming projects during the course time while teaching the Computer Programming course, and the teacher cited local examples to explain the concept to students. Computer Programming teacher always showed a keen interest in motivating students to accomplish Computer Programming practical's, also Computer Programming teacher didn't explain beyond the examples in the textbooks as he/she was using.

The findings from the results revealed that teachers did programming successfully in class during teaching computer programming in universities. 12.3% strongly agreed, 62.3% agreed, 14.9% undecided, 9.6% disagreed and .9% strongly disagreed. The weighted average of 3.75 (4.5 >WA≥ 3.5) indicated high confidence in university teachers' did programming successfully in class. It also revealed that 15.8% strongly agreed, 48.2% agreed, 27.2% were undecided, and 8.8% disagreed and 0% strongly disagreed. The weighted average of 3.71 (4.5 >WA≥ 3.5) indicated high confidence, teacher gave Computer Programming projects during the course time and was identified as the major challenges but complete the tasks. According to the results 12.3% strongly agreed, 50.0% agreed, 29.8% were undecided, 7.9% disagreed and 0% strongly disagreed. The weighted average of 3.67 (4.5 >WA≥ 3.5) indicated high confidence, while teaching the CP course, the teacher cited local examples to explain the concept to students as stated and was identified as the key factors which influenced the students to successfully understand the concepts.

(ii) In the middle-level challenges, the teacher felt insufficient in testing and evaluation

Regarding the findings indicated that teachers felt insufficient in testing and evaluation and identified as the major challenges faced by teachers during teaching computer programming in universities were 20.2% strongly agreed, 41.2% agreed, 20.2% undecided, 13.2% disagreed and 5.3% strongly disagreed. The weighted average of 3.58 (4.5 >WA≥ 3.5) indicated high confidence of teachers who feel insufficient in testing and evaluation.

(iii) In the lower-level challenges, the teacher felt insufficient in testing and evaluation

As indicated by the outcomes, it uncovered that Computer Programming instructor was consistently accessible for inquiries after classes if understudies had not perceived talk unmistakably and distinguished as the significant difficulties looked by educators during training PC programming in colleges were 9.6% strongly agreed, 50.9% agreed, 14.9% undecided, 21.9% disagreed and 1.8% strongly disagreed oppose this idea. The weighted normal of 3.45 (3.5 >WA≥ 2.5) demonstrated high certainty of the Computer Programming educator was consistently accessible for inquiries after classes if understudies had not perceived the talk unmistakably. In response to the question of the supply course outline and plan at the beginning of the semester, the responses revealed that according to 69.3% teachers supplied course outline and had to use in their daily learning and following the schedule as planned and 30.7% did not get supplied course outline and plan at the beginning of the semester. From the results, it indicated that the question of specified the objectives of every new topic was responded 68.4% of teachers specify the objectives of every new topic and 31.6% did not specified the objectives of every new topic at the beginning of the semester and this showed that most teachers specified the objectives of every new topic as a result, their learning was not difficult due to the facts that teachers should specify the objectives of every new topic. Furthermore, 76.3% of teachers made a brief review of the previous lecture before starting a new lecture to make the lessons interesting and motivating while 23.7% of teachers did not make a brief review of the previous lecture before starting a new lecture to make the lessons interesting and motivating.

According to the result, the question of organised and presented the lecture in a logical sequence (simple to complex; known to the unknown; theory to practice), responses reveal that 86.0% of teachers state it's organised and presented the lecture in a logical sequence (simple to complex; known to the unknown; theory to practice) while 14.0% teachers responses state that wasn't organised and presented the lecture in a logical sequence (simple to complex; known to the unknown; theory to practice) and in general, it was well organised from simple to complex. From the results the question of delivered and communicated the information and concept articulately and effectively, responses revealed that 77.2% of teachers state it was well delivered and communicated the information and concept articulately and effectively while 22.8% of teachers responses stated that wasn't delivered and communicated the information and concept articulately and effectively. It was responded that 78.1% of teachers' opinions encouraged students to participate in classes, while 21.9% of teacher's responses did not encourage students to participate in class and in general, conclusion basing on the highest response that encouraged students to participate in classes.

Based on the item from it, responses revealed that 82.5% of teachers use these teaching methods (lecturing, demonstration, group discussion, student involvement, etc.) were effective and useful, while 17.5% of teachers did not use teaching methods which include (lecturing, demonstration, group discussion, student involvement, etc.) were effective and useful hence the highest response encourages them to ensure that they use these teaching methods. Regarding the findings, responses indicated that 86.8% of teachers used teaching aids while teaching programming course, while 12.3% of teachers did not use teaching aids while teaching programming course and hence the highest response used the teaching aids while teaching the programming courses. The responses revealed that 89.1% of teachers used projectors during lectures of programming courses,

while 10.9% of teachers did not use projectors during lectures of programming courses and hence the highest response used projectors during lectures of the programming courses.

According to the contents of the responses revealed that 78.1% of teachers teach in well-organised classrooms, while 21.2% of teachers did not teach in well-organised classrooms and hence the highest response of teachers used to teach in well-organised classrooms. From the findings, the responses indicated that 79.8% of teachers can manage the classroom effectively when delivering a lecture regarding programming language by using a computer or laptop, while 20.2% of teachers did not manage the classroom effectively when delivering lectures regarding programming language by using computer or laptop. The results responded, 89.5% of teacher needs enough skills when teaching programming language by using appropriate tools, while 10.5% of teachers didn't need enough skills when teaching programming language by using appropriate tools and hence response with the highest percentage of teachers needs enough skills when teaching programming language by using appropriate tools. The findings and recommendations of this study were of great importance to the teachers on how to teach and improve on major challenges faced by teachers during teaching computer programming courses in universities. The study was of great importance for policymakers, lecturers to help them with pedagogical techniques used in teaching computer programming in universities.

The following recommendations should be put into consideration to play a vital role to advance and improve the skills in computer programming courses in universities.

1 The government should introduce programming course basics as a subject at the higher secondary level to allow the science students to prepare for the programming courses at the colleges and which will be helpful in universities.

2 The universities should use high-speed internet service providers (ISP), to allow the teachers and learners to use the internet.

3 The universities should think of introducing a human-computer interaction (HCI) web applications, where a student can visit the university HCI web application and they can be able to interact with the system directly without a teacher in place anywhere a student he/she is, for practicing more content, and live coding, and debugging online as tools to easy computer programming.

5 Limitations of the study

The discoveries of this investigation ought to be deciphered with an alert since there are restrictions related to the study design. The information of this investigation was gathered from the 12 universities of Uganda by convenience sampling. To sum up the discoveries, further investigations ought to be completed across various nations and should assess the variety in the segment and financial attributes among members. The consequences of this study are restricted to the scope of exactness and reasonableness of the instruments utilised in this investigation.

6 Conclusion

Computer Programming is not difficult but interesting if someone knows how to deal with this. However, teachers have different challenges to use computer programming effectively in the classroom. Teachers must teach Computer Programming by themselves hence practicing constructivist learning by admitting and involving learners for the understanding of programming-related activities which will deal through hands-on practice. Furthermore, to be a success in computer programming, one should be able to change the abstract problems to a working solution on the computer by doing program coding. For novices, they need to set up learning procedures to help them to deal with the cycle of computer programming as well as to influence the way the novice identifies the programming process for them to succeed in programming. The survey identified different major challenges faced by teachers during teaching computer programming courses. The results provide the general amount of data on the observed challenges related to programming concepts and program construction. The survey also provided information on teacher's opinions of the most useful learning situations and equipment types to be used to improve computer programming courses. These results can be used when coming up with approaches and designing materials suitable for basic computer programming courses. Teachers should be able to visualise the computer programming state during code execution and are one of the tests that teachers face during teaching computer programming courses. However, the visualisation tool is used during reasonable exercises in the research center through gathering action with the goal that instructors and students can talk about and investigate all effectively the exercises embraced. Further research may be conduct based on qualitative data and may relate between quantitative and qualitative findings.

References

Agustin, M., Lourdes, S. and Montebon, D.R.T. (2018) 'An assessment of project teacher exchange for ASEAN teachers (TEACH) program', *International Journal of Evaluation and Research in Education*, Vol. 7, No. 1, pp.1–10.

Alghamdi, M.Y.A. (2017) *Supporting the learning of computer programming in an early years education*, Doctoral dissertation, Liverpool John Moores University.

Amin, M.Y.M. and Rahimi, A. (2018) 'Challenges faced by novice EFL teachers', *International Journal of Humanities and Cultural Studies*, Vol. 5, No. 1.

Andreoli, J.M., Pareschi, R. and Castagnetti, T. (1997) 'Static analysis of linear logic programming', *New Generation Computing*, Vol. 15, No. 4, pp.449–481.

Angeli, C. and Valanides, N. (2009) 'Epistemological and methodological issues for the conceptualization, development, and assessment of ICT-TPCK: advances in technological pedagogical content knowledge (TPCK)', *Computers and Education*, Vol. 52, No. 1, pp.154–168. https://doi.org/10.1016/j.compedu.2008.07.006

Carbonaro, A. (2019) 'Good practices to influence engagement and learning outcomes on a traditional introductory programming course', *Interactive Learning Environments*, Vol. 27, No. 7, pp.919–926.

Čisar, S.M., Pinter, R. and Čisar, P. (2014) 'Code hunt—"hunting" to learn programming', *2014 IEEE 15th international symposium on computational intelligence and informatics (CINTI)* (pp.329–333). IEEE.

Gauch, H.G. and Moran, D.R. (2019) 'AMMISOFT for AMMI analysis with best practices', *BioRxiv*, 538454.

Ghavifekr, S., Kunjappan, T., Ramasamy, L. and Anthony, A. (2016) 'Teaching and learning with ICT tools: issues and challenges from teachers' perceptions', *Malaysian Online Journal of Educational Technology*, Vol. 4, No. 2, pp.38–57.

Gomes, A. and Mendes, A.J. (2007) 'Learning to program-difficulties and solutions', *International Conference on Engineering Education–ICEE*, Vol. 7.

Hooshyar, D., Ahmad, R.B., Nasir, M.H.N.M., Shamshirband, S. and Horng, S.J. (2015a) 'Flowchart-based programming environments for improving comprehension and problem-solving skill of novice programmers: a survey', *International Journal of Advanced Intelligence Paradigms*, Vol. 7, No. 1, pp.24–56.

Hooshyar, D., Ahmad, R.B., Yousefi, M., Yusop, F.D. and Horng, S.J. (2015b) 'A flowchart-based intelligent tutoring system for improving problem-solving skills of novice programmers', *Journal of Computer Assisted Learning*, Vol. 31, No. 4, pp.345–361.

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J.M. and Irwin, J. (1997) 'Aspect-oriented programming', *European conference on object-oriented programming* (pp.220–242). Springer, Berlin, Heidelberg.

Koryuhina, C. and Shamshina, T. (2018) 'Challenges of ICT in education. In The 16th international scientific conference information technologies and management. ISMA University. Available online at: https://www.isma.lv/FILES/SCIENCE/IT&M2018_THESES/03_MDM/15_IT &M2018_Korjuhina_Shamshina. pdf.

Martínez, L.G., Martínez, L.G., Licea, G., Juárez, J.R. and Aguilar, L. (2014) 'Experiences using PSP and XP to support teaching in undergraduate programming courses', *Computer Applications in Engineering Education*, Vol. 22, No. 3, pp.563–569.

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y.B.D. et al. (2001) 'A multi-national, multi-institutional study of assessment of programming skills of first-year CS students', *Working group reports from ITiCSE on Innovation and technology in computer science education*, pp.125–180.

Nawrocki, J.R., Walter, B. and Wojciechowski, A. (2002) 'Comparison of CMM level 2 and eXtreme programming', *European Conference on Software Quality* (pp. 288-297). Springer, Berlin, Heidelberg.

Oroma, J.O., Wanga, H. and Ngumbuke, F. (2012) 'Challenges of teaching and learning computer programming in developing countries: Lessons from Tumaini university', *Proceedings of INTED2012 conference. 5th–7th march 2012, Valencia, Spain*.

Papadakis, S. (2018) 'Is pair programming more effective than solo programming for secondary education novice programmers? A case study', *International Journal of Web-Based Learning and Teaching Technologies (IJWLTT)*, Vol. 13, No. 1, pp.1–16.

Papadakis, S. (2020) 'Evaluating a game-development approach to teach introductory programming concepts in secondary education', *International Journal of Technology Enhanced Learning*, Vol. 12, No. 2, pp.127–145.

Papadakis, S. and Kalogiannakis, M. (2017) 'Using gamification for supporting an introductory programming course. The case of classcraft in a secondary education classroom', *Interactivity, game creation, design, learning, and innovation* (pp.366–375). Springer, Cham.

Papadakis, S. and Orfanakis, V. (2018) 'Comparing novice programing environments for use in secondary education: App Inventor for Android vs. Alice', *International Journal of Technology Enhanced Learning*, Vol. 10, Nos. 1/2, pp.44–72.

Pelgrum, W.J. (2001) 'Obstacles to the integration of ICT in education: results from a worldwide educational assessment', *Computers & education*, Vol. 37, No. 2, pp.163–178.

Petri, G. and von Wangenheim, C.G. (2017) 'How games for computing education are evaluated? A systematic literature review', *Computers & Education*, Vol. 107, pp.68–90.

Rahman, N., Nandigam, D. and Tirumala, S.S. (2015) *Teaching computer programming with a coaching mindset*.

Suryani, A. (2008) Comparing Case Study and Ethnography as Qualitative Research Approaches', *Journal Ilmu Komunikasi*, Vol. 5, No. 1, pp.117–127. https://doi.org/10.24002/jik.v5i1.221

Teague, D. and Lister, R. (2014) 'Longitudinal think aloud study of a novice programmer', *Conferences in Research and Practice in Information Technology Series*.

Teague, D., Fidge, C. and Xu, Y. (2016) 'Combining unsupervised and invigilated assessment of introductory programming', *Proceedings of the Australasian Computer Science Week Multiconference* (pp.1–10).

Tshabalala, T. and Ncube, C. (2014) 'Teachers' perceptions on challenges faced by rural secondary schools in the implementation of the technical and vocational education and training policy in Nkayi district', *International Research Journal of Teacher Education*, Vol. 1, No. 2, pp.10–15.

Van den Akker, M., Brinkkemper, S., Diepen, G. and Versendaal, J. (2004) 'Flexible Release Composition using Integer Linear Programming', *Technical Report*, *03018* (UU-CS 2004-063).

Van den Beemt, A. and Diepstraten, I. (2016) 'Teacher perspectives on ICT: A learning ecology approach', *Computers & Education*, Vol. 92, pp.161–170. https://doi.org/10.1016/j.compedu .2015.10.017

Vihavainen, A., Paksula, M. and Luukkainen, M. (2011) 'Extreme apprenticeship method in teaching programming for beginners', *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education - SIGCSE '11*, 93. https://doi.org/10.1145/1953163.1953196