ISSN 2513-8359

# The perceptions of secondary school students regarding the learning of computer programming at Key Stage 4 in the revised curriculum in England: difficulties and prospects.

**Almabrok Musa ALMDAHEM**
a.almadahem@wlv.ac.uk
Faculty of Science and Engineering, School of Computer Science
University of Wolverhampton, England, UK

**DOI: 10.21585/ijcses.v7i1.147**

**Abstract**

A national curriculum for the study of computing became compulsory in English secondary schools in September 2014, replacing the study of information and communications technology with computer science (CS). This posed difficulties for teachers and students who did not have knowledge or experience of programming. This study was designed to investigate and gain a critical understanding of the teaching of computer programming (CP) at Key Stage 4 (KS4; year 10 - 11) of the CS curriculum, including assessing the impact of learning CP and students' perceptions of CS and their overall performance in the subject. Furthermore, the study investigated the measures to improve the teaching of CP and the factors that have an impact on the effective teaching of the CP curriculum. The study sample comprised 300 students. The findings indicate that the main difficulties the study found that the issues faced by students learning programming include a lack of time, the perceptions that it is a 'difficult' subject and students' insufficient understanding of programming. The findings also suggest that schools have made efforts to overcome these challenges and are willing to adopt programming as a subject and to help, encourage, develop and improve students' ability to learn programming; however, the results indicate that it is essential that schools address the shortage of teaching staff with specialised knowledge of CP. This study revealed that three factors can help to overcome the difficulties where the three factors are for students (perceptions towards learning and teaching programming, benefits, and support). The findings of this study will be useful for students who are learning programming in secondary schools.

**Keywords:** programming, National Curriculum, Key Stage 4(KS4), secondary school, difficulties, and learning.

## 1. Introduction

A 2012 report by the Royal Society advocated replacing the existing information and communications technology (ICT) curriculum in England with a wider-ranging subject to be known as 'computing' (The Royal Society, 2012). In 2014, the Department for Education (DfE) replaced England's national curriculum for ICT for secondary schools with a revised computing curriculum (Moller and Crick, 2018). Computing is now a compulsory part of the national curriculum for schools and provides important learning opportunities. The revised computing curriculum has three strands: computer science (CS); digital literacy (DL); and information technology (IT) (Lau, 2017) at Key Stage 4 (KS4; year 10 - 11) (see Figure 1). As computers are becoming an inseparable part of everyday life, the need and demand for computer programming (CP) is increasing rapidly. England was one of the first countries to take the initiative to integrate CS into its school curriculum (Passey, 2017). CS is the study of both software and hardware design, including principles of information processing and how digital systems work. In the CS element of the curriculum, students are taught the basic principles of programming, how digital systems work, and how to put this knowledge to use through programming. For many students, programming is regarded as one of the most challenging aspects of CS. Several computing education researchers have sought to establish the causes of students' programming difficulties and have identified the lack of knowledge as one of the contributors (Sentance, Waite, and Kallia, 2019). Computer programming is becoming increasingly important to many societies around the world, and is a skill required by most educational institutions. However, the teaching of programming is not well developed in many secondary schools. Today, the teaching of programming is considered to be a priority in several countries; hence the interest in this domain and the extent to which research in the field is growing. The revised national curriculum for computing was introduced by the Department for Education (DfE) in England in September 2014 with the intention of providing students with the necessary skills and knowledge in this area of study (Larke, 2019). It replaced the national curriculum for ICT in secondary schools (Moller and Crick, 2018). The 2014 revised curriculum included the application of mathematical skills, such as abstraction, decomposition (divide the problem to small parts to be manageable and easier to understand), logic, algorithms, and data representation. As computing is a key curriculum subject in all types of schools, this shift requires support for teachers to encourage new knowledge; teachers and students should therefore be given clear guidance on using computers successfully to support the teaching and learning of the subject (De Paula, Valente and Burn, 2014). This change brought with it a number of challenges. Prior to the introduction of the revised curriculum, ICT was often limited to the development of media, office-type software, and exploration of web-based resources (Woollard, 2017). The terms ‘computer science and ‘programming’ are used in the revised curriculum; however, these words are not interchangeable, and these terms are defined in Chapter 2. According to the Royal Society (2019), there are 3,954 teachers of computing and 8,834 ICT teachers. RSA Oxford, Cambridge (OCR) found that there were 50,605 CS students in 2018. There are 24,323 schools in England, of which 3,448 are secondary schools (British Educational Suppliers Association; BESA, 2019).

1. What are KS4 students’ perceptions of the learning of programming?

This paper investigated students’ perceptions of the learning of programming. (see page 2)

2. Are there ways in which the teaching of CP can be enhanced? (see page 11)

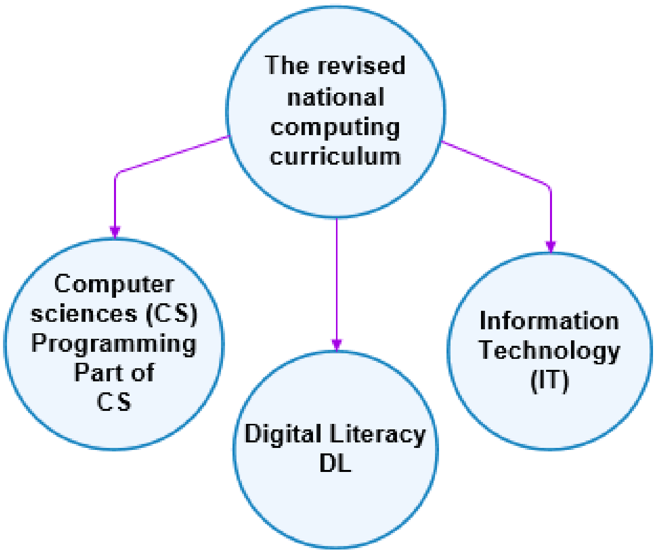


Figure 1. The national curriculum’s computing programme of study (Lau, 2017, p.4).

## 2. Literature review

### *2.1 Significance of the study*

This study addresses the way that the revised computing curriculum handles programming instruction, and it assesses the factors affecting effective implementation of the curriculum as well as the learning of programming. It also seeks to understand students’ perceptions of this topic. Learning CS involves learning programming; therefore, it is important to develop students’ fluency in this area of study. To become competent in programming, students need to be able to comprehend the concepts and use them well. Although this may seem to require extra time and effort, it is central to the learning of CS. Moreover, the original contribution of this study lies in the discussion related to the implementation of the revised CS curriculum in England’s secondary schools. For a pedagogical model to be successful, it is important to examine its construction and effectiveness of application in secondary schools. This study contributes to the literature by considering students’ opinions of such challenges for future improvement of CS/CP learning.

### *2.2 Students’ perception*

One of the difficulties faced by students when learning programming is the lack of lesson time. Students aged 14 years typically have one hour per week of computing lessons (The Royal Society, 2017). A recent survey showed that 30% of secondary schools reported a decrease in the total time allocated to teaching CS, while 22% saw an increase (Royal Society, 2017). As mentioned by 40% of the surveyed secondary schools that only provide one hour a week or fortnight for lessons in computing for 11–14-year-olds, teachers do not have sufficient time to ensure that the subject is adequately covered; therefore, enough time needs to be allocated in school timetables for covering the three strands of the computing curriculum (The Royal Society, 2017). A reduction in teaching hours would also make it more difficult for teachers to gain enough experience and confidence to teach programming (The Royal Society, 2017).

An additional issue that has an effect on students’ learning is being taught by teachers who are not subject specialists in programming; in fact, when a specialist teacher is absent from class, students' performance can decline, especially if there is no teacher with the same expertise to cover the absent teacher’s role (Ost and Schiman, 2017). Some studies show that the numerous problems encountered in learning programming lead to the misunderstanding of the concepts of programming. For students, these issues may become obstacles to gaining programming skills, and these and other similar factors affect their progress in learning programming (Yukselturk and Altiok, 2017).

Another issue is students’ mathematical skills. Duran (2016) tested the hypothesis that excellent mathematical skills yield excellent academic performance in CS; programming involves solving problems by applying mathematics and using computers for calculations. The ability to programme can simply be transferred to the performance of mathematical tasks. The development of the knowledge and understanding of programming will equip students with the creativity and skills to use a variety of new technologies.

### *2.3 Learning programming*

People are already living in a world controlled by software, which is why it is so important for children to learn the basic elements of programming. For example, television is delivered over the internet; telephone calls are transmitted over software-controlled networks; people do not buy maps anymore but use the web; medical care is delivered online; and people buy their goods by shopping online. The next generation’s world will be even more online and digital (Crow, 2014). Programming was once thought to be a task reserved for computer scientists, but in the twenty-first century, it is seen as a crucial and required talent that everyone should master (Shim, Kwon and Lee, 2016). Educational systems around the world are encouraging students to engage in programming activities and develop their programming skills (Scherer, Siddiq and Sánchez Viveros, 2019). High-quality teaching and learning of programming can help students to meet the digital challenges of the 21st century (Yildiz Durak, 2018). Programming is believed to help students succeed in other subjects in school, and it positively impacts on students’ future employment prospects. It is a beneficial educational activity that helps students to develop and improve in other skills, such as problem-solving, and boost critical thinking and logical reasoning skills (Kalelioglu, and Gülbahar, 2014). Programming has a positive effect on high school students’ reasoning skills and self-efficacy problem-solving in mathematics. The learning of programming will also improve students’ skills in other subjects and creativity and enhance collaboration. In fact, collaboration between students will considerably improve individual programming skills by reducing the frustration experienced by students and increasing their enjoyment and satisfaction in learning programming. Students will also be better prepared to collaborate as a group, for example, in pair programming. The retention of students in CS courses will also be improved (Li, Plaue and Kraemer, 2013).

In the future, students who have programming knowledge will be able to be innovative and solve problems more effectively, with fewer obstacles to impede their success. There is no doubt that the study of programming is beneficial to all students in their everyday life; the benefits also extend to positive impacts on processing, thinking, and communication (Jancheski, 2017). Sáez-López, Gonzalez and Cano (2016) argued that programming lessons can be beneficial to school students and that they should be engaged in programming, The value and success of applying visual programming from active methodologies education are highlighted by a grasp of computational ideas, project-based learning, active approach, usefulness, and commitment and motivation. This was confirmed by Laylaec (2019); there is no doubt that learning programming can be beneficial to students in the future, relating, for example, to improving employment and academic study opportunities. Taylor, Vasquez and Donehower, (2017) stated that as technology became more personalized, students should be given the opportunity to study programming, as this knowledge will equip them well for the future. Students who have programming knowledge and skills may be able to solve many of the problems of society by using their know-how of computer technologies and inventive notions in the future, encouraging students to learn motivation and understanding for programming by training and collaborative learning is important (Hayashi, Fukamachi and Komatsugawa, 2015). According to

Psycharis and Kallia (2017) learning programming may also provide many benefits for students' cognitive skills which can be applied to a variety of subjects, this indicates that learning programming can help students develop skills that can be applied to other subjects such as mathematics, science, and engineering.

Programming is increasingly considered to be a significant skill in modern societies and, as Nager and Atkinson (2016) highlighted, it can lead to many employment opportunities, whilst it is a fundamental skill featured in the revised national curriculum for computing. Some research shows that the study of programming is becoming less prevalent, and recently the number of students choosing to study CS courses has also because of difficulties experienced in gaining CS skills (Azmi, Iahad and Ahmad,2015). Such difficulties include the fundamental concepts of programming, for example, construction loops, structure control and algorithms (Eltegani and Butgereit, 2015). However, one of the most difficult issues in education is how to keep students motivated, encouraged and interested in the learning (Eltegani and Butgereit, 2015). For students, learning programming is onerous because it requires considerable work, dedication, and training. The difficulties of learning programming are a cause for concern everywhere where this subject is needed (Vahldick, Mendes and Marcelino, 2014; Figueiredo and García-Peñalvo, 2018(. These difficulties include, for example, lack of resources, such as textbooks for students; students' difficulties in understanding programming concepts; the lack of experience of the teacher giving programming lessons; students finding the lessons 'boring'; and insufficient teaching and learning time (Sentance and Csizmadia, 2017). On the other hand, motivation and interest are significant variables in the learning of programming. The lack of these attributes will push many students to give up CP; therefore, several studies have been carried out in an effort to improve students' motivation and interest in programming (Shim, Kwon and Lee,2016). Further study is still needed to uncover the different problems encountered in learning this subject; students frequently experience difficulties in grasping basic and essential concepts of programming, leading to disappointment and confusion (Galgouranas, and Xinogalos, 2018). Despite this increasing lack of interest in CS among students (Combéfis, Beresnevičius and Dagienė, 2016), the NCCE revealed in 2019 that it would open 23 new computing centres across England, providing assistance, including teaching support and resources, to teachers of secondary computing (Snowdon, 2019).

### *2.4 Student support*

Support and motivation for learning programming in school are important for students. According to Kafai and Burke (2015), the use of educational games in the classroom is beginning to be seen as promising because of the evidence that such games can increase student performance and motivation. Motivation has a significant role in academic achievement: higher motivation can result in increased academic achievement. Because the learning of programming necessitates constant practice, maintaining students' motivation is of the highest importance. Students need stronger support to stimulate them to become engaged in learning activities as well as support for collaboration to motivate and improve different models of teamwork (Khaleel, Ashaari, Wook, and Ismail,2017; Santos, Gomes,and Mendes, 2010). Teachers also have a large role in supporting their students; this is especially the case for teachers who have greater experience in programming and can support students by performing a variety of activities and demonstrating skills in the field of programming.

According to Alsubaie (2016), teachers have a responsibility to develop appropriate instructional strategies to help schools achieve curriculum objectives, as well as developing suitable approaches to students' learning. Moreover, teachers should support students in improving the skills needed for success in all their courses. It is also important to enhance the learning environment of students to offer a world-class computing education (O'Kane, 2019). Providing effective support or guidance is the key to the improvement of students' performance (Yang, Hwang, Yang, and Hwang, 2015). It is essential to provide students with good programming tools as practice facilitates the study of programming (Kazimoglu, Kiernan, Bacon, and MacKinnon, 2012). There is a big role for teachers in supporting students, both from a motivational standpoint and from a pedagogical standpoint. It may be worth considering ways to track and boost students' motivation and self-confidence. Specifically, appropriate instructional and pedagogic techniques will increase students' motivation, self-confidence, and perceptions of competence, thus increasing their willingness to put in the effort necessary to learn how to program. Further, learning what influences programming favourably and unfavourably can help students overcome the inherent challenges of learning programming by creating the ideal learning environment and pedagogical approaches. So, motivation is crucial in learning programming. As a result, any instructional strategy should include motivating student techniques; this is especially true in courses where a very active student attitude is fundamental (Gomes et al., 2018).

### *2.5 Difficulties in learning programming*

It is known that many students have difficulty learning programming, especially its concepts; a number of studies have shown that students may not have sufficiently developed skills and the knowledge to start learning programming (Wang et al., 2017). The learning of programming can be considered as an iterative operation. In the beginning, the student is taught simple and basic information and where to apply it. Students need to grasp fundamental programming concepts (for example, repetition, sequence, condition, branch, variable and function) and learn the use of instructions and syntax for programming and tools (Moons and De Backer, 2013; Shim, Kwon and Lee, 2016). According to Yukselturk and Altiok (2017), some students can have trouble obtaining the requisite competencies while studying programming so that lessons become challenging. However, there is evidence in the literature that some of the major difficulties of learning programming are ineffective learning, lack of interest in programming and lack of motivation to study programming (Khaleel et al., 2017). Students encounter several problems, for example, misunderstanding of programming and lack of resources and time. Despite these views that programming is difficult to learn, Luxton-Reilly (2016) states that it is actually easy to learn and that, with little effort, almost anyone can learn programming; all that learners need to do is collectively shift their mindset and reach achievable goals. Thus, it is important to recognise and study new teaching methods that focus on students' learning and ameliorating difficulties, consequently resulting in students' active involvement in learning (Piteira, Costa and Aparicio, 2018).

## 3. Method

### *3.1 Research design*

The methodology used in this study is based on a mixed-methods approach, which usually involves gathering, analysing, and integrating data collected from qualitative approaches, such as open-ended, and quantitative data from surveys. Data were collected from secondary school students through a questionnaire and then an analysis of the collected data was undertaken. The last stage was the interpretation and analysis of the data. Based on the nature of the sample used in this study, the questionnaire was created for KS4 secondary school students. The questionnaires designed for students included four factors: students' perception; students' learning; support; and difficulties.

### *3.2 Participants*

The participants in this research were secondary school students. Responses for the questionnaire came 32 schools and it was proposed that all the Year 10 and 11 students involved in the fill-in were asked to complete questionnaires to ascertain their perceptions about the learning of programming. The aim and objectives of the research were explained, including the use of questionnaires and interviews. Altogether, a total of 300 students were selected to complete questionnaires and about 10 students were selected for participation in open-ended questions. There were both female and male students aged 15-16 years old, and the data was collected in secondary schools in the several schools in England.

### *3.3 Ethical approval*

According to the UK Data Protection Act (1998/2018), anyone processing, obtaining, holding or disclosing personal data must comply with the data protection principles. Personal data include sensitive information such as factual information about and personal opinions of the individual. The participants of this study were informed that their personal data would be processed in accordance with the rights of data subjects and would be destroyed after the project. It was also explained that the data would be protected from unauthorised or unlawful processing and would not be transferred to another country. Moreover, the researcher secured the personal data of the participants by not recording names on the questionnaires or using names in the presentation of the findings of the study. The researcher's role in the research process was explained to the students. The ethical guidelines of the British Educational Research Association (1992) were adhered to throughout this study. These guidelines emphasise respect for persons, knowledge, democratic values, and quality in educational research.

#### *3.3.1. Ethical considerations of research with children*

Dealing with children implies the need for a degree of the right to respect. It also underlines the importance of carefully protecting children's rights throughout the study process (Pillay, 2014). Researchers should ensure that children are not harmed in any way through their participation in research (Broström, 2006).

#### *3.3.2. Obtaining consent*

For the child participants who were under the age of 18 years old in this study, approval was obtained from parents or legal guardians. According to Heath et al. (2007), obtaining informed consent from parents / guardians of children is vital to the ethical research process. Since children are frequently less familiar with what research necessitates, they may initially wish to participate but later feel less keen as they get to know what is involved. As a result, consideration should be given to how children might be made to feel at ease with terminating their participation in the research if they so desire.

### *3.3.3. Confidentiality, anonymity, and safeguarding*

In keeping with the topic of this research, stringent ethical measures were taken throughout the research process. The study's ethical approval was obtained by the Ethics Committee of the Faculty of Science and Engineering

Anonymity, confidentiality and safeguarding are ethical procedures designed to protect the privacy of human subjects while collecting and analysing. As part of this study, no children have been mentioned by name and the results from individual students cannot be attributed to a single school in the presented results.

### *3.3.4. Respect*

This research gave all students the right to express their views about their own experiences in their life of study. In addition, participants in this research are seen as indispensable and worthy partners in research. The outcomes of the study were therefore achieved by promotion, protection, and respect of the rights of students are made intrinsic to every stage and level of research. The collection of participants' views and ideas (students' perceptions in this study) about a social phenomenon (the teaching of programming) seems to be a valuable way to generate credibility and gain trustworthiness and respect.

## 4. Data analysis

The quantitative analysis of the data collected from the survey questionnaires completed by 300 students who participated in the study. The analysis focused on students' perceptions of learning programming, as well as the challenges encountered in learning/teaching programming in secondary schools. To investigate and gain a critical understanding of the learning of programming in the KS4 computing curriculum, Spearman's rank-order correlation coefficients and multiple linear regressions were used to determine the following: for students, whether there was a relationship between difficulties experienced when learning programming and the perceptions of learning programming, benefits of learning programming, and the teaching support provided at school for programming.

### *4.1 Quantitative findings*

Table 1. What are the perceptions of KS4 students of programming in the revised curriculum?

| | **Male (N = 160)** | | | | | | **Female (N = 140)** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Factor (1) perceptions of teaching programming / Effect factors on teaching programming** | | | | | | | | | | | |
| Q6 | My positive perception of learning programming helps me study better. | | | | | | | | | | | |
| | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) |
| | 21 (13.1) | 22 (13.8) | 20 (12.5) | 74 (46.3) | 23 (14.4) | 3.4 (1.3) | 13 (9.3) | 16 (11.4) | 19 (13.6) | 70 (50.0) | 22 (15.7) | 3.5 (1.2) |
| Q7 | When I find all the necessary resources and good teachers at school, I am motivated to study programming. | | | | | | | | | | | |
| | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) |
| | 10 (6.3) | 26 (16.3) | 16 (10.0) | 49 (30.6) | 59 (36.9) | 3.8 (1.3) | 12 (8.6) | 35 (25.0) | 19 (13.6) | 36 (25.7) | 38 (27.1) | 3.4 (1.3) |
| Q8 | Competition in learning programming with my classmates pushes me to perform better. | | | | | | | | | | | |
| | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) |
| | 2 (1.3) | 9 (5.6) | 28 (17.5) | 89 (55.6) | 32 (20.0) | 3.9 (0.8) | 5 (3.6) | 5 (3.6) | 16 (11.4) | 83 (59.3) | 31 (22.1) | 3.9 (0.9) |
| Q9 | The pressure from a teacher and my classmates forces me to learn to programme better and work harder. | | | | | | | | | | | |
| | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) |
| | 3 (1.9) | 6 (3.8) | 21 (13.1) | 70 (43.8) | 60 (37.5) | 4.1 (0.9) | 3 (2.1) | 2 (1.4) | 12 (8.6) | 73 (52.1) | 50 (35.7) | 4.2 (0.8) |
| Q10 | When my classmates do better, I am motivated to study harder to keep up | | | | | | | | | | | |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) |
| | 3 (1.9) | 4 (2.5) | 19 (11.9) | 99 (61.9) | 35 (21.9) | 4.0 (0.8) | 3 (2.1) | 5 (3.6) | 22 (15.7) | 78 (55.7) | 32 (22.9) | 3.9 (0.8) |
| **Factor (2) Benefits of learning programming** | | | | | | | | | | | | |
| Q11 | Learning programming in secondary school will enhance my confidence in this subject in the future. | | | | | | | | | | | |
| | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) |
| | 4 (2.5) | 5 (3.1) | 16 (10.0) | 87 (54.4) | 48 (30.0) | 4.1 (0.9) | 3 (2.1) | 8 (5.7) | 14 (10.0) | 77 (55.0) | 38 (27.1) | 4.0 (0.9) |
| Q12 | Learning programming in secondary school will improve my performance in other subjects | | | | | | | | | | | |
| | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) |
| | 30 (18.8) | 8 (5.0) | 9 (5.6) | 110 (68.8) | 3 (1.9) | 3.3 (1.2) | 28 (20.0) | 6 (4.3) | 14 (10.6) | 83 (59.3) | 9 (6.4) | 3.3 (1.3) |
| Q13 | Learning programming in secondary school will make it easier to complete studies in this subject at university level. | | | | | | | | | | | |
| | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) |
| | 3 (1.9) | 9 (5.6) | 16 (10.0) | 79 (49.4) | 53 (33.1) | 4.1 (0.9) | 5 (3.6) | 4 (2.9) | 20 (14.3) | 74 (52.9) | 37 (26.4) | 4.0 (0.9) |
| Q14 | Learning programming in secondary school gives me a chance to continue learning to programme in the future. | | | | | | | | | | | |
| | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) |
| | 4 (2.5) | 5 (3.1) | 22 (13.8) | 79 (49.4) | 50 (31.3) | 4.0 (0.9) | 3 (2.1) | 8 (5.7) | 30 (21.4) | 65 (46.4) | 34 (24.3) | 3.9 (0.9) |
| **Factor (3) Difficulties experienced when learning programming** | | | | | | | | | | | | |
| Q15 | Learning to programming at secondary school will motivate me to study it in the future. | | | | | | | | | | | |
| | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) |
| | 4 (2.5) | 7 (4.4) | 20 (12.5) | 96 (60.0) | 33 (20.6) | 3.9 (0.9) | 5 (3.6) | 4 (2.9) | 29 (20.7) | 84 (60.0) | 18 (12.9) | 3.8 (0.8) |
| Q16 | Learning programming is difficult. | | | | | | | | | | | |
| | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) |
| | 3 (1.9) | 4 (2.5) | 10 (6.3) | 78 (48.8) | 65 (40.6) | 4.2 (0.8) | 2 (1.4) | 3 (2.1) | 17 (12.1) | 77 (55.0) | 41 (29.3) | 4.1 (0.8) |
| Q17 | Challenging programming exercises motivate me to work harder. | | | | | | | | | | | |
| | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) |
| | 2 (1.3) | 19 (11.9) | 14 (8.8) | 73 (45.6) | 52 (32.4) | 4.0 (1.0) | 1 (0.7) | 17 (12.1) | 10 (7.1) | 73 (52.1) | 39 (27.9) | 3.9 (0.9) |
| **Factor (4) Teaching support provided at school for programming** | | | | | | | | | | | | |
| Q18 | Support for teaching computer programming is good in secondary school. | | | | | | | | | | | |
| | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) | S.D (%) | D (%) | N (%) | A (%) | S.A (%) | M (SD) |
| | 0 (0.0) | 5 (3.1) | 24 (15.0) | 109 (68.1) | 22 (13.8) | 3.9 (0.6) | 0 (0.0) | 10 (7.1) | 20 (14.3) | 97 (69.3) | 13 (9.3) | 3.8 (0.7) |

Note. S.D = strongly disagree D disagree; N = neither disagree nor agree; S.A = strongly agree A= agree; M = mean; SD = standard deviation; % = percent

Table 2. Descriptive statistics and Cronbach's alpha analysis of the factors affecting students 'learning of programming.

| | Factor | Items | M | SD | Cronbach's alpha |
|---|---|---|---|---|---|
| Students | Perception of learning programming | 5 | 3.80 | 0.47 | 0.830 |
| | Benefits of learning programming | 4 | 3.82 | 0.54 | 0.780 |
| | Difficulties experienced when learning programming | 3 | 3.99 | 0.66 | 0.704 |
| | Teaching support provided at school for programming | 1 | 3.87 | 0.67 | NA |

Note: M = mean; SD = standard deviation; NA = not applicable.

To determine whether there was a relationship between difficulties experienced by students when learning programming and their perception of learning programming, benefits of learning programming, and teaching support provided at school for programming, Spearman's rank-order correlation tests and multiple linear regression were employed. According to the results of Spearman's rank-order correlation tests, there was a significantly negative relationship between the score for difficulties experienced when learning programming and those for perception of learning programming (rs = -0.182, p = 0.002), benefits of learning programming (rs = -0.345, p < 0.001), and teaching support provided at school for programming (rs = -0.331, p < 0.001).

Table 3. Spearman's rank-order correlation coefficients for the relationship between the score for challenges experienced by students when learning programming and those for the three factors of interest.

| **Factor of interest** | **Correlation coefficient (*p*-value)** |
|---|---|
| Perception of learning programming | -0.182 (0.002) |
| Benefits of learning programming | -0.345 (< 0.001) |
| Teaching support provided at school for programming | -0.331 (< 0.001) |

The results of the multiple linear regression (Table 4) show that the predictor, the score for perception of learning programming, contributed significantly to the model (t(96) = -2.728, p = 0.007). The relationship between the score for difficulties experienced when learning programming and that for perception of learning programming was significantly negative (B = -0.199, SE = 0.073). That is, students perceiving learning programming more positively experienced less difficulty when learning programming. Based on these results, $H0_{1s}$ (no relationship between the score for difficulties experienced when learning programming and that for perception of learning programming) was rejected as the regression coefficient was significantly negative, with a *p*-value of < 0.05. Therefore, it was concluded that the more positive the students' perception of learning programming was, the less the difficulty they experienced when learning programming. The predictor, the score for benefits of learning programming, contributed statistically significantly to the model (*t* (96) = -6.472, *p* < 0.001). There was a significantly negative relationship between the score for difficulties experienced when learning programming and that for benefits of learning programming (B = -0.426, SE = 0.066). That is, students perceiving greater benefits of learning programming experienced less difficulty when learning programming. Based on these results, $H0_{2s}$ (no relationship between the score for difficulties experienced when learning programming and that for benefits of learning programming) was rejected as the regression coefficient was significantly negative, with a *p*-value < 0.05. Therefore, it was concluded that the greater the students' perception of benefits of learning programming, the less difficulty they experienced when learning programming. The predictor, the score for teaching support provided at school for programming, contributed statistically significantly to the model (*t*(96) = -5.234, *p* < 0.001). The relationship between the score for difficulties experienced when learning programming and that for teaching support provided at school for programming was negative (B = -0.264, SE = 0.050). That is, students who had a higher perception of teaching support provided at school for programming experienced less difficulty when learning programming. Based on these results, $H0_{3s}$ (no relationship between the score for difficulties experienced when learning programming and that for teaching support provided at school for programming) was rejected as the regression coefficient was negative, with a *p*-value value of < 0.05. Therefore, it was concluded that the greater the students' perception of teaching support provided at school for programming, the less difficulty they experienced when learning programming.

Table 1. Results of the multiple linear regression for determining the relationships between the score for difficulties experienced by students when learning programming and those for three factors of interest.

| **Factor** | **B** | **SE** | **t** | **p** | **VIF** |
|---|---|---|---|---|---|
| Constant | 5.026 | 0.098 | 51.047 | < 0.001 | |
| Perception of learning programming | -0.199 | 0.073 | -2.728 | 0.007 | 1.189 |
| Benefits of learning programming | -0.426 | 0.066 | -6.472 | < 0.001 | 1.267 |

| | | | | | |
|---|---|---|---|---|---|
| Teaching support provided at school for programming | -0.264 | 0.050 | -5.234 | < 0.001 | 1.152 |

Note B = parameter estimate, SE = standard error, $t$ = $t$-statistic, $p$ = $p$-value, and VIF = variance inflation factor.

Summary of the findings of the quantitative analysis aimed to investigate students' perceptions of learning and programming. The results indicate that students had a positive perceptions of learning programming and perceived it as beneficial. However, they experienced high levels of difficulty when learning programming but believed that teaching support provided at school for studying programming was good. To determine whether there was a relationship between difficulties experienced by students when learning programming and their perceptions of learning programming, benefits of learning programming, and teaching support provided at school for programming, Spearman's rank-order correlation tests and multiple linear regression were employed. According to the results of Spearman's rank-order correlation tests, there was a significantly negative relationship between the score for difficulties experienced when learning programming and those for perceptions of learning programming, benefits of learning programming, and teaching support provided at school for programming. Similarly, based on the results of the multiple linear regression, there was a statistically significantly negative relationship between the score for difficulties experienced when learning programming and those for perceptions of learning programming, benefits of learning programming, and teaching support provided at school for programming.

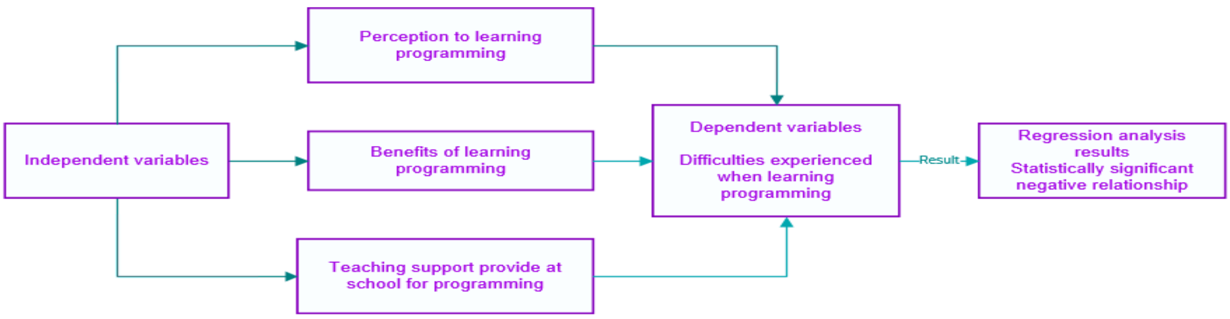


Figure 2. Regression analysis of data for students.

The results indicated that students with a more positive perception towards learning programming and perceived greater benefits of learning programming and greater teaching support provided at school for programming experienced less difficulty when learning programming (Figure 2). Cronbach's alpha was computed for items under each factor to determine the reliability of the construct. The results of Cronbach's alpha analysis indicated a high reliability of the construct for students. The quantitative results indicated that students learning programming difficulties lie in a lack of time, lack of content programming lack of experiences lack of programming knowledge, lack of motivation the perception that it is a 'difficult' subject, and students' insufficient understanding of programming. Students believed that when they perceived more support, motivation, good resources, and skilful programming teachers, and more activities, they would be more interested and motivated to learn programming. The quantitative results were significantly negative, which suggests that for students the difficulties in learning programming would be overcome by an increase in support, benefits, and perceptions.

*4.2 Qualitative findings*

This section discussed the responses garnered through open-ended surveys of students. It provides important explanations for the learning of programming. The results of the qualitative analysis of research question 1 confirmed that students, both males and females, received support from their schools to study programming and that programming clearly helped students in their learning of other subjects. The responses to the open-ended questions provided some evidence that the learning of programming enhanced future opportunities for higher education and careers. There were no gender differences in these responses. In addition, the findings for research question 1 show that most of the respondents noted and confirmed the benefits of introducing programming into the Key Stage 4 curriculum on the basis that programming is useful and that it is an important skill necessary for students in the future. The responses to research question 2 reveal that the development of students' confidence in studying CP would facilitate the learning of complicated topics in communication processes and practices of programming. Having presented the analysis of the relevant data of this study.

## 5. Discussion

### *5.1 Research question 1: What are KS4 students' and teachers' perceptions of the learning of programming?*

#### *5.1.1 Students' perceptions*

In order to discuss the data generated from this research question, this section starts with presenting the students' perceptions. One of the main knowledge areas within the CS curriculum is programming, which includes algorithms, concepts, patterns, programming paradigms and technologies (Halim and Phon, 2020). The results of the quantitative analysis indicate that when students are provided by suitable classroom environment, resources, and skilful programming teachers, they will be interested and motivated to learn programming. In addition, students believe that learning programming enhances their confidence in future studies and improves their achievements in other subjects, and they feel that practical learning support leads them to perform better. However, many students believe that they need to learn programming, but they realise that the learning of programming is not an easy task. Practical support is a significant component of programming courses, and it is an important process for students to develop their skills. 'Practice is considered an important step in grasping the precise concepts of computer programming for novices' (Malik, 2016, p.1).

Programming is often actively linked to learning in other subjects such as mathematics, science, and technology (Otterborn, Schönborn and Hultén, (2020). Programming helps students acquire skills that are prerequisites for success in other subjects, including problem-solving, critical thinking, creativity, collaboration, mathematical thinking, and reasoning (Psycharis and Kallia, 2017; Tsai, 2019; Partovi, 2020). Learning programming in secondary schools affects students both in the moment and in the future. Saez-Lopez et al. (2020) reported that programming knowledge offers advantages and benefits related to various fields by providing the skills that stimulate motivation and digital competence. LópezLeiva et al. (2022) noted that students enjoyed using programming while applying mathematics to develop images and videos that they chose and created through programming. The findings of this study indicate that students considered that programming can help them in their educational career. In the context of this study, the majority of students believe that learning programming is a useful skill that will help them later in life. In addition, some students indicated that the learning of programming is a vital skill that all students must acquire. However, some students are not interested to learn programming, even though they were aware of the benefits. Thus, although future plans can be a contributing factor to choosing to learn CS, it is not the only factor to consider.

The revised computing curriculum was developed to provide young people with the computing knowledge, understanding, and foundational skills they require now and will require in the future (Dredge, 2014). The data obtained from the survey and the open-ended questions in this study showed that students and teachers both believe that programming education will help students in other subjects in school. However, some students may not be interested in the future study of programming at college or university level, or they may only take a short course after graduation from school and get a job, while other students who like programming in secondary school may not want to continue with it at university and instead, they tend to study other subjects.

In this study, students acknowledged that a good teacher is one of the significant elements of teaching programming. The teacher has a crucial role in the education process in class and school. Teachers are the primary source of knowledge and that can positively impact on the students' achievement. Some of the students indicated that the absence of a good teacher affects their learning of programming, although it is likely that they were referring to a direct effect through their teacher's absence or illness. This study indicates that in the event that there is no subject teacher, it is important to supply the class with a dedicated teacher for CS or programming. The study also provides evidence that experienced teachers will have a direct effect on students' success (Gage et al., 2018). Moreover, the absence of a teacher specialising in the field has an effect on students' learning, especially when the replacement does not meet the same requirement (Ost and Schiman, 2017). It is important to provide support and professional development opportunities for those who are currently teaching computing in schools (Moller and Powell, 2019). This is a major reason for guiding and helping students to improve in their study of programming. The results of this study showed that students with a more positive attitude to learning programming experienced fewer difficulties when learning the subject. This study also showed that a large percentage of students did not have a computer at home. The absence of a home computer may lead to students' lack of experience, confidence, and time for learning programming in their school. The results of this study showed that the majority of students do not own a computer at home, which was perceived to have a negative impact on their skills at programming. This finding is consistent with that of Fairlie and Robinson (2013) in that when students do have access to a home computer, they can gain experience, confidence, and the time to devote to programming. Advantages of home computers include helping students with their learning, students' increased desire to create resources and artefacts, development of more skills, improvement of their existing skills, greater experience, and increased flexibility in the times when they can use computers (Fairlie, 2012). However, home computer use also has disadvantages, including the considerable amount of time spent playing games which leads to students having no energy or time for their studies. (Fairlie and London, 2012). Some students noted that they were not confident in learning programming and that this had an impact on their interest and motivation to learn the subject. This was also explained by Shim, Kwon and Lee (2016) who stated that many students see the field of programming as a difficult subject and students feel disappointed when they do not make the progress that they believe they should. Consequently, programming difficulty had an effect on students' decisions either to choose CS or not. Studying programming needs a significant amount of knowledge, skills, time, and practice, which does not inspire students who are looking to study CS as an option at school (Benjamin, 2017). Therefore, several scholars conclude that programming is a complicated process and there are challenges and problems in teaching and learning the subject (for example, Prasad and Chaudhary, 2021). However, maybe a lack of programming capacity is not the only reason why some students decide not to select studying programming, and there may be other reasons such as a lack of interest or motivation. Other important factors are students' poor grasp of programming content and lack of experience, which can make the learning of programming difficult. This is consistent with the fact that the study of programming requires inspiration, knowledge, ability, skills, time, and practice (Benjamin, 2017).

Another issue that makes programming difficult is mathematics. Research conducted by Mozelius, Ulfenborg and Persson (2019), showed that the lack of knowledge and mathematical skills makes programming a difficult subject; however, they believe that programming should have a positive effect on students' mathematical skills. This study concurred with Duran (2016) who tested the hypothesis that excellent mathematical skills yield excellent academic performance in CS, as programming is solving problems by applying mathematics. Programming is an important skill necessary for mathematics and sometimes programming failure rates can be, partially, attributed to a lack of mathematical capacity.

*5.1.2 Are there ways in which the teaching of CP can be enhanced?*

The result of this study shows that students believe that collaborative programming is important and an effective approach to learning. According to Bravo, Duque and Gallardo (2013), collaborative programming increases confidence and enhances the value of learning programming. Collaborative interactions in learning programming, such as pair programming, can develop more positive feelings and experiences than individual programming (Cal and Can, 2020). Identification of the factors that affect the achievements and confidence of students using pair programming can enable teachers and curriculum developers to make better decisions on the use of this approach in secondary school programming courses (Cal and Can, 2020). It should be noted, however, that collaborative learning is a complex method that entails the co-creation of knowledge (Tsan et al., 2021) and that further research is required to determine the impact of such pedagogic approaches. Demir and Seferoglu (2021), for example, noted that only a small number of studies have experimentally demonstrated that pair programming is effective. The issue of on-going professional development was mentioned by some of the teachers, including more support with different pedagogical approaches that could help to stimulate students in the classroom environment.

The use of games was mentioned as a possibility. Some studies such as Papadakis (2020) mentioned that the game development and programming environment approach has a positive effect on students' motivation and achievement of basic programming skills in CS lessons. Moreover, since robots are real tools that assist students to understand the concepts of programming, the use of programme robots will enable students to enter a potentially fun and appealing learning environment (Alalawi and Said, 2020). Research conducted by Dlab et al. (2020) showed that the use of modern learning methods and appropriate digital content and tools, including games, is more effective for achieving teaching and learning goals. This research confirmed that the learning of programming through the use of games promotes good programming practices and enables students to understand the concepts of programming. Several studies have focused on the issue of students' motivation for learning programming (for example, Zarei et al., 2020). High-performance computing artefacts provide students with opportunities to increase their understanding and improve their learning of CS (Mwasaga and Joy, 2020). Learning through educational games can contribute positively to the learning outcomes and increase the students' motivation in the learning programming (Mathew, Malik and Tawafak, 2019). Motivation plays a significant role in learning programming; it assists the students in learning the basic concepts of programming. Moreover, these games enhance competition and collaboration between students. These results broaden the knowledge and help to overcome the difficulties faced by students in the learning of programming in secondary schools and could shed light on future policymaking in relation to curriculum development for secondary schools.

## 6. Recommendations

Recommendation regarding the perceptions of students of programming

1) This study recommends that schools should motivate and support students to give them confidence in themselves to succeed in this field.

2) Schools should consider increasing the teaching time allocated to programming lessons by providing more extracurricular activities, for example, after-school programming clubs.

3) Schools should provide appropriate technical resources to support the teaching of programming.

4) Collaboration is important for students; it is a significant technique for developing higher-quality learning and is recommended as a pedagogical aid for CS/programming teachers (this study suggests pair programming as an important support for students).

5) Schools should consider providing high-performance computing artefacts (tools) to increase students' understanding, and improve their learning of CS.

6) In the case of students who do not have a home computer, schools should cooperate with parents to provide a computer or laptop to support and assist them in learning programming or any other subject.

**References**

Alalawi, W.A. and Said, M.N.H.B.M. (2020) Using Lego mindstorms robotics programming in enhancing computational thinking among middle school in Saudi Arabia. *International Journal of Psychosocial Rehabilitation*, 24(05), pp. 5367–5373.

Alsubaie, M. A. (2016). Curriculum development: Teacher involvement in curriculum development. *Journal of Education and Practice*, *7*(9), 106-107.

Azmi, S., Iahad, N.A. and Ahmad, N. (2015) Gamification in online collaborative learning for programming courses: A literature review. *ARPN Journal of Engineering and Applied Sciences*, 10(23), pp. 1-3.

Benjamin Wohl (2017) *Coding the curriculum: new computer science GCSE fails to make the grade*[online][Accessed 5 January 2019]. available at: Coding the curriculum: new computer science GCSE fails to make the grade | The Independent | The Independent

Bravo, C., Duque, R. and Gallardo, J. (2013) A groupware system to support collaborative programming: Design and experiences. *Journal of Systems and Software*, 86(7), pp. 1759–1771.

Broström, S. (2006) Children's perspectives on their childhood experiences. Nordic childhoods and early education: Philosophy, research, policy, and practice in Denmark, Finland, Iceland, Norway, and Sweden, pp. 223-255.

Cal, H. and Can, G. (2020) The Influence of Pair Programming on Secondary School Students' Confidence and Achievement in Computer Programming. *Trakya Eğitim Dergisi,* 10(1), pp. 221-237.

Combéfis, S., Beresnevičius, G. and Dagienė, V. (2016) Learning programming through games and contests: Overview, characterisation and discussion. *Olympiads in Informatics*, 10(1), pp. 39–60.

Crow, D. (2014) Why every child should learn to code [online]? [Accessed 12 September 2018]. Available at: https://www.theguardian.com/technology/2014/feb/07/year-of-code-dan -crow-songkick

Demir, O. and Seferoglu, S.S. (2021) A comparison of solo and pair programming in terms of flow experience, coding quality, and coding achievement. *Journal of Educational Computing Research*, 58(8), pp. 1448–1466.

Dlab, M.H., Hoic-Bozic, N., Mezak, J. and Zunic, M., (2020) Supporting Croatian Primary School Teachers in Designing Game Based Learning Activities: A Case Study Martina Holenko Dlab , Natasa Hoic-Bozic , Jasminka Mezak , and Marina Zunic University of Rijeka, Department of Informatics, Rijeka, Croatia University of Rijeka, Faculty of Teacher Education. *ECGBL 2020 14th European Conference on Game-Based Learning (p. 125).* Academic Conferences limited.

Dredge, S. (2014) *Coding at school: A parent's guide to England's new computing curriculum. The Guardian* [online]. [Accessed 16 November 2017]. Available at: https://www.theguardian.com/technology/2014/sep/04/coding-school-computing-children-programming

Duran, I. L. (2016). The role of mathematics background in the performance of BSCS students in computer programming subject. *International Journal of Multidisciplinary Research and Modern Education (IJMRME)*, *2*(1), 147-50.

Eltegani, N. and Butgereit, L. (2015) Attributes of students' engagement in fundamental programming learning. *International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering* (ICCNEEE). IEEE, pp. 101–106.

Fairlie, R. W. (2012). The effects of home access to technology on computer skills: Evidence from a field experiment. *Information Economics and Policy*, *24*(3-4), 243-253.

Fairlie, R. W., & London, R. A. (2012). The effects of home computers on educational outcomes: Evidence from a field experiment with community college students. *The Economic Journal*, *122*(561), 727-753.

Fairlie, R. W., & Robinson, J. (2013). Experimental evidence on the effects of home computers on academic achievement among schoolchildren. *American Economic Journal: Applied Economics*, *5*(3), 211-40.

Figueiredo, J. and García-Peñalvo, F.J. (2018) Building skills in introductory programming. In: *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality,* pp. 46–50.

Gage, N. A., Scott, T., Hirn, R., & MacSuga-Gage, A. S. (2018). The relationship between teachers' implementation of classroom management practices and student behavior in elementary school. *Behavioral disorders*, *43*(2), 302-315.

Galgouranas, S. and Xinogalos, S. (2018) JAVANT-GARDE: A cross-platform serious game for an introduction to programming with Java. *Simulation & Gaming,* 49(6), pp. 751-767.

Gomes, A., Ke, W., Lam, C.T., Marcelino, M.J. and Mendes, A. (2018) Student motivation towards learning to program. In *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1-8).

Jancheski, M. (2017) Improving teaching and learning computer programming in schools through educational software. *Olympiads in Informatics*, 11, pp. 55–75.

Halim, N. F. A., & Phon, D. N. E. (2020, Februar). Mobile Learning Application Impact Towards Student Performance in Programming Subject. In *IOP Conference Series: Materials Science and Engineering* (Vol. 769, No. 1, p. 012056). IOP Publishing.

Hayashi, Y., Fukamachi, K.I. and Komatsugawa, H. (2015) Collaborative learning in computer programming courses that adopted the flipped classroom. In: *International Conference on Learning and Teaching in Computing and Engineering*. IEEE, pp. 209–212.

Heath, S., Charles, V., Crow, G. and Wiles, R., (2007) Informed consent, gatekeepers and go-betweens: negotiating consent in child and youth-orientated institutions. *British Educational Research Journal,* 33(3), pp. 403-417.

Kafai, Y.B. and Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational psychologist*, *50*(4), 313-334. *https://doi. org/10.1080/00461520.2015.1124022.*

Kalelioglu, F., & Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education*, *13*(1), 33-50.

Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, *9*, 522-531.

Khaleel, F. L., Ashaari, N. S., Tengku, T. S. M., & Ismail, A. (2017). Programming learning requirements based on multi perspectives. *International Journal of Electrical and Computer Engineering*, *7*(3), 1299.

Larke, L.R. (2019) Agentic neglect: Teachers as gatekeepers of England's national computing curriculum. *British Journal of Educational Technology,* 50(3), pp. 1137-1150.

Lau, W. (2017). Teaching Computing in Secondary Schools: A Practical Handbook. Routledge.

laylaec (2019) Advantages and disadvantages of programming education [online].

[Accessed: 2 August 2019]. Available at: Advantages and disadvantages of programming education (laylaec.com)

Laylaec (2019) *Advantages and disadvantages of programming education* [online]. [Accessed: 2 August 2019]. Available at: https://laylaec.com/2019/04/29/advantages-and-disadvantages-of-programming-education/

Li, Z., Plaue, C., & Kraemer, E. (2013, May). A spirit of camaraderie: The impact of pair programming on retention. In *2013 26th International Conference on Software Engineering Education and Training (CSEE&T)* (pp. 209-218). IEEE.

LópezLeiva, C.A., Noriega, G., Celedón-Pattichis, S. and Pattichis, M.S. (2022) From students to cofacilitators: Latinx students' experiences in mathematics and computer programming. Teachers College Record, 124(5), pp. 146-165.

Luxton-Reilly, A. (2016, July). Learning to program is easy. In Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (pp. 284-289).

Malik, S.I., (2016) Enhancing practice and achievement in introductory programming using an ADRI editor. In: *2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pp. 32-39. IEEE.

Mathew, R., Malik, S.I. and Tawafak, R.M. (2019) Teaching Problem Solving Skills using an Educational Game in a Computer Programming Course. *Informatics in education,* 18(2), pp. 359-373.

Moller, F., & Crick, T. (2018). A university-based model for supporting computer science curriculum. Reform Journal of Computers in Education, 5(4), 415-434.

Moller, F., & Powell, S. (2019, January). Teaching Computing via a School Placement. In *Proceedings of the 3rd Conference on Computing Education Practice* (pp. 1-4).

Mozelius, P., Ulfenborg, M., & Persson, N. (2019). Teacher attitudes towards the integration of programming in middle school mathematics. *INTED 2019*, 701-706.

Moons, J., & De Backer, C. (2013). The design and pilot evaluation of an interactive learning environment for introductory programming influenced by cognitive load theory and constructivism. *Computers & Education*, *60*(1), 368-384.

Mwasaga, N.M. and Joy, M., (2020) Using high-performance computing artifacts as a learning intervention: A systematic literature review. *Proceedings of the 2nd International Conference on Intelligent and Innovative Computing Applications.* pp. 1–10.

Nager, A. and Atkinson, R.D. (2016) *The case for improving US computer science education* [online]. [Accessed: 12 January 2019]. Available at: https://www2.itif.org/2016-computer-science-education.pdf

Nikiforos, S., Kontomaris, C., & Chorianopoulos, K. (2013). MIT scratch: A powerful tool for improving teaching of programming. *Conferrence on Informatics in Education*, 1-5.

O'Kane, L. (2019) *Computing mastery for primary schools: Achieving computing mastery* [online]. Accessed: 2 May 2019]. Available at: computing mastery Archives - iCompute (icompute-uk.com)

Ost, B., & Schiman, J. C. (2017). Workload and teacher absence. *Economics of Education Review*, *57*, 20-30.

Otterborn, A., Schönborn, K.J. and Hultén, M. (2020) Investigating preschool educators' implementation of computer programming in their teaching practice. *Early Childhood Education Journal,* 48(3), pp. 253-262.

Papadakis, S. (2020) Evaluating a game-development approach to teach introductory programming concepts in secondary education. *International Journal of Technology Enhanced Learning*, 12(2), pp. 127–145.

Passey, D. (2017). Computer science (CS) in the compulsory education curriculum: Implications for future research. *Education and Information Technologies*, *22*(2), 421-443.

Pillay, J., (2014) Ethical considerations in educational research involving children: Implications for educational researchers in South Africa. *South African Journal of Childhood Education,* 4(2), pp. 194-212.

Piteira, M., Costa, C., & Aparicio, M. (2018). Computer programming learning: how to apply gamification on online courses?. *Computer programming learning: how to apply gamification on online courses?*, (2).

Prasad, A., & Chaudhary, K. (2021). Interactive Animation and Affective Teaching and Learning in Programming Courses. In *Advances in Computer, Communication and Computational Sciences* (pp. 613-623). Springer, Singapore.

Psycharis, S. and Kallia, M. (2017) The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science*, 45(5), pp. 583–602.

The Royal Society (2012) *Shutdown or restart? The way forward for computing in UK schools* [online]. [Accessed 19 April 2017]. Available at: https://royalsociety.org/-/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf

Royal Society, *After the reboot: computing education in UK schools* [Online], (2017). [Accessed 12 September 2018]. Available at: <https://royalsociety.org/-/media/policy/projects/computing-education/computing-education-report.pdf/>

Sáez-López, J.M., Román-González, M. and Vázquez-Cano, E. (2016) Visual programming languages integrated across the curriculum in elementary school: A two-year case study using "Scratch" in five schools. *Computers & Education*, 97, pp. 129–141.

Sáez-López, J.M., del Olmo-Muñoz, J., González-Calero, J.A. and Cózar-Gutiérrez, R. (2020) Exploring the Effect of Training in Visual Block Programming for Preservice Teachers. *Multimodal Technologies and Interaction*, 4(3), pp. 2-10.

Santos, Á., Gomes, A., & Mendes, A. J. (2010). Integrating new technologies and existing tools to promote programming learning. *Algorithms*, *3*(2), 183-196.

Scherer, R., Siddiq, F. and Sánchez Viveros, B. (2019) The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. Journal of Educational Psychology, 111(5), pp. 764–792.

Sentance, S. and Csizmadia, A. (2017) Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, 22(2), pp. 469–495.

Sentance, S. (2019). Moving to mainstream: developing computing for all. In *Proceedings of the 14th Workshop in Primary and Secondary Computing Education* (pp. 1-2).

Shim, J., Kwon, D., & Lee, W. (2016). The effects of a robot game environment on computer programming education for elementary school students. *IEEE Transactions on Education*, *60*(2), 164-172.

Snowdon, K. (2019) *First 23 computing hubs in England announced* [online]. [Accessed 12 October 2019]. Available at: https://schoolsweek.co.uk/computing-hubs-england-announced-ncce/

Taylor, M.S., Vasquez, E. and Donehower, C., (2017) Computer programming with early elementary students with Down syndrome. *Journal of Special Education Technology*, 32(3), pp. 149–159.

Tsai, C. Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. *Computers in Human Behavior*, *95*, 224-232.

Tsan, J., Vandenberg, J., Zakaria, Z., Boulden, D.C., Lynch, C., Wiebe, E. and Boyer, K.E. (2021) Collaborative Dialogue and Types of Conflict: An Analysis of Pair Programming Interactions between Upper Elementary Students. In: *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pp. 1184-1190.

Wang, X. M., Hwang, G. J., Liang, Z. Y., & Wang, H. Y. (2017). Enhancing students' computer programming performances, critical thinking awareness and attitudes towards programming: An online peer-assessment attempt. *Journal of Educational Technology & Society*, *20*(4), 58-68.

Woollard, J. (2017) Computer studies, information technology, ICT and now computing. In *Debates in Computing and ict Education*, pp. 14-26. Routledge.

Vahldick, A., Mendes, A.J. and Marcelino, M.J. (2014) A review of games designed to improve introductory computer programming competencies. In *2014 IEEE frontiers in education conference (FIE) proceedings* (pp. 1-7).

Yang, T.C., Hwang, G.J., Yang, S.J. and Hwang, G.H. (2015) A two-tier test-based approach to improving students' computer-programming skills in a web-based learning environment. *Journal of Educational Technology & Society*, 18(1), pp. 198–210

Yildiz Durak, H. (2018). Digital story design activities used for teaching programming effect on learning of programming concepts, programming self-efficacy, and participation and analysis of student experiences. *Journal of Computer Assisted Learning*, *34*(6), 740-752.

Yukselturk, E., & Altiok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, *48*(3), 789-801.

Zarei, A., Mohd-Yusof, K., Kolivand, H., Ahmadi, A. and Al-Hamar, Y. (2020) A new game-based strategy for enhancing youth programming skills. *Journal of Internet Technology*, 21(5), pp. 1289–1301.