

Article

Development of Computational Thinking Using Microcontrollers Integrated into OOP (Object-Oriented Programming)

Melinda Timea Fülöp ^{1,*} , József Udvaros ², Ákos Gubán ² and Ágnes Sándor ²¹ Faculty of Economics and Business Administration, Babeş-Bolyai University, 400591 Cluj-Napoca, Romania² Faculty of Finance and Accountancy, Budapest Business School, 1149 Budapest, Hungary; udvaros.jozsef@uni-bge.hu (J.U.); guban.akos@uni-bge.hu (Á.G.); sandor.agnes@uni-bge.hu (Á.S.)

* Correspondence: melinda.fulop@econ.ubbcluj.ro

Abstract: Nowadays, the theme of computer thinking is a common topic for educational research. The scientific literature on the subject has gradually appeared, in which psychologists emphasize the need for the development of thinking of children. Research often relates only to the development of computational thinking at elementary and high schools. Nowadays, almost everything is digitalized, so it is important to also develop the computational thinking skills of students at higher levels. In our study, we present the results of the conducted data analysis in which we examined the development of students' efficiency. On the basis of the results, we propose a possible solution for the development of computational thinking. Using data research processes, we examined the results of the object-oriented (OO) planning and programming subject of Budapest Business School, going back 5 years. The results show that the level of particular computational thinking could be measured using the exam results, and teacher advancement (experience) improved the level of particular computational thinking. Today, education has been greatly influenced by COVID-19, challenging not only teachers but also students. The production of particular computational thinking under COVID-19 or online is much more effective than the pursuit of full computational thinking through traditional teaching.

Keywords: particular computational thinking; microcontrollers; programming

Citation: Fülöp, M.T.; Udvaros, J.; Gubán, Á.; Sándor, Á. Development of Computational Thinking Using Microcontrollers Integrated into OOP (Object-Oriented Programming). *Sustainability* **2022**, *14*, 7218. <https://doi.org/10.3390/su14127218>

Academic Editors: Antonio Messeni Petruzzelli and Maxim A. Dulebenets

Received: 20 April 2022

Accepted: 10 June 2022

Published: 13 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the 21st century, the role of information in society has significantly grown. The ability to obtain information has become valuable. To perform our day-to-day work, we need to obtain the information we need to fully integrate the opportunities offered by information technology into our daily lives. In addition to computers and mobile phones, other smart devices have also appeared and have an impact on our lives. It is important that we take advantage of these new tools to enjoy the opportunities of the new information society. The growing demand for the use of smart devices has led to the introduction of information, communication and technology (ICT) literacy in the initial onset of personal development [1–3]. The information society affects all areas of our lives. This change does not only affect the economy, but the whole social system is being rebuilt based on the logic of the information society. Moreover, we are an integral part of that [4]. In the information society, learning and teaching methodologies constantly change as a result of ICT tools. This is true for both public and higher education. People's daily lives are greatly influenced by digital technology, for instance, entertainment, leisure activities, obtaining and passing on information, keeping in touch, etc. Digital languages must be learnt to be successful in the digital world, with problem solving through coding and computer thinking as operational paradigms [5]. Based on the above, education information technology (IT) professionals and researchers address the question of what changes are taking place in the field of

learning and teaching through digital technology and how learning and teaching can be effective using these tools. What teaching methods should be used to achieve the goal?

The aim of our study is to present the results of a data analysis in which the development of students' success in the case of OOP subjects was examined. Furthermore, based on the results and the literature analysis, we propose a possible solution for the development of computational thinking, with which we believe we can improve the performance of students. In the course of the literature analysis, we focused on the development of computational thinking and the results of the solutions (research studies) that support it. We examined whether particular computational thinking depends on the number of lessons. Our research questions are outlined below.

What is the meaning of computational thinking?

What is the meaning of particular computational thinking?

How did COVID-19 affect the invention of particular computational thinking?

What are the cornerstones of computational thinking, and what are their dimensions?

What tools and methods are used in the international literature to develop computational thinking?

What are the exam results examined? What conclusions can we draw from them? How can we achieve better results?

Based on statistics, in this modern society, employees with the best practical digital skills can easily prevail in the labor market, but employers are also searching for employees who not only are able to use but also develop smart devices well. Because of this, the importance of the teaching and training of digital skills and competencies increases.

Aware of the importance of eSkills, which includes related information technology and digital skill terms, many articles exist on the problems of teaching programming in primary education [6–9]. According to Weintrop and his colleagues, the main elements are code-literacy skills for science, technology, engineering and mathematics (STEM subjects) [10]. Nowadays, developing computational thinking abilities is a very popular topic among researchers (teachers). It is important to provide examples for teachers and to support STEM activities with the implementation of computational thinking [11,12].

Lifelong learning is very important, as recommended by the EU, because environmental sustainability is a multidisciplinary and rapidly evolving field of education, with the population as a whole at its heart. Lifelong learning should, therefore, be pursued in environmental sustainability education, and the cooperation between the education and training sectors should be strengthened. Interdisciplinary and participatory learning and teaching, as well as collaboration, are needed to equip learners with the competences needed for sustainability. Linking teaching and learning, the physical (learning) environment, partnerships and the community are ideal for teaching environmental sustainability, because they bring the subject closer to the real world. Several member states use this whole-school, community-based approach instead of the classroom methodology, but it has not yet been established in the EU.

Online education is in a paradoxical situation in terms of sustainability. The additional emissions associated with contact hours, such as traveling to the educational institution, attendance, energy consumed, lighting, heating–cooling, etc., are individually precipitated to a much lesser extent. There are no unnecessary elements—no empty, unused classrooms—as these emissions do not occur. The individual equipment used by the students can be integrated into education, and there is no need to purchase additional equipment, no need to mix and match. These operations can be very damaging from a sustainability point of view, as well as from a recycling point of view. Due to the increase in energy use and device utilization, sustainability is much higher, thus the pursuit of online individual knowledge in any case results in lower emissions (CO₂ equivalent emissions). It can, therefore, greatly improve the sustainability aspect.

2. Methodology

In the first step of the analysis, data filtering was performed. As the data were available in the Institution's internal ERP system (NEPTUN), the first task was to extract the required records from the available database with tens of years of data. We could not use a well-defined method because we did not have the structure of the database system and built-in complex filtering tool. Therefore, a hierarchical heuristic filtering was performed [13]. Neptun records all data throughout the students' academic lifecycle. Not only grades but also logins, course enrollments, etc. In our case, this was the basis of the study. This system has several filtering options. We applied the filters to the period, subject and grades. These data are quite representative, as 33% of the business informatics training in Hungary during this period was covered by the respective Budapest Business School institution. Therefore, the data can be considered representative. The data cleaning was performed by the software system itself through the data cleaning function of the Neptun system. The data received were already cleaned and filtered. In other words, we conducted a complete query for the years under review, from which we automatically deleted the fields necessary to identify the students; we conducted this using our own small manual-tuning software. Based on the anonymous records of the order of 100,000, we created a processing database that had to be further filtered. Then, in the next step, we had to find what could identify the object within the given records, and we also had to determine which encoding, in the case of different encodings (the code could change from year to year), could be considered the same. Filtering was easy from this point on. The results of filtering clearly determined all the exam results related to the OO planning and programming subject of the Budapest Business School for the last 5 years. Thus, we obtained a database of 3296 records, whose structure was explored and was considered suitable as input to the IBM SPSS Statistics application. Then, in the last step, we performed a simple frequency study and searched for a fit to the distributions [14]. After the tests, the obtained results were evaluated.

In the literature review, we used the following keywords and phrases in English in the databases of Web of Science, Scopus and ScienceDirect to analyze the topic of computer thinking: "computational thinking", "new teaching methods", "teaching methods", "microcontrollers in education", "algorithmical thinking" and "developing of computational thinking".

The search covered the period from 2005 to 2021, and it was important for ICT and education that studies older than 5–10 years were not considered due to the rapid development of technology. Due to the topicality of the topic, we tried to examine the studies only dating back five or six years, but we considered it necessary to mention the results of a few older studies that we considered to be determinant ones in the topic. After analyzing the titles and abstracts of the articles, 40 studies remained that were related to our article. In general, most of the articles were about the use and results of different electronic devices (microcontrollers, robots) in the educational process. Different courses were presented as the teaching of programming is practice-oriented. After a thorough reading of the articles, 5 studies relevant to the research study were identified, which dealt specifically with computer thinking.

In the course of our research study, we could define three hypotheses.

H1. *The level of particular computational thinking can be measured using the exam results.*

H2. *Teacher advancement (experience) improves the level of particular computational thinking.*

H3. *The production of particular computational thinking during COVID-19 or online is much more effective than the pursuit of full computational thinking through traditional education.*

3. Results

Computational thinking has already been defined by many and has mostly been related to critical thinking, problem solving and creativity.

There are several definitions for determining computational thinking, some of which are here reported. In the article [15], it is written that computational thinking is about

“problem solving, systems design and understanding human behaviour based on basic concepts in computer science where a new perspective on the relationship(s) between people and computers and leads to a wave of computational thinking research.”

Berland and Wilensky defined computational thinking as “the ability to think with the computer as a tool” and suggested using “computational perspectives” as an alternative to “computational thinking” to emphasize that computational thinking can be constrained by contexts [16].

In 2011, Wing gave the following definition: “Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be carried out by an information-processing agent” [17].

Brennan and Resnick classified the components of computational thinking into three dimensions:

1. Computational concepts that we use during programming;
2. Computational practices, which are practices that we use during problem solving;
3. Computational perspectives that are the students’ understanding of the digital world [18].

Wing said that computational thinking has four cornerstones (key techniques): decomposition, pattern recognition, abstraction and algorithms. Decomposition means dividing a difficult problem into simple problems. The pattern recognition term describes searching for similarities among the individual parts of the complex problem. Abstraction means concentration only for the main information. Algorithms describe the method of a problem [15].

Juškevičienė and Dagienė, in their article, conduct a literature review about the main components of computational thinking and define computational thinking abilities. These abilities are necessary in various topics, for example, programming, problem solving or constructionism [19]. Juškevičienė, in another article, clarified the computational thinking abilities and defined eight dimensions:

1. Data analysis and representation;
2. Computing artefacts;
3. Decomposition;
4. Abstraction;
5. Algorithms;
6. Communication and collaboration;
7. Computing and society;
8. Evaluation [11,12].

Shute, Sun and Asbell-Clarke dealt with the differences between computational thinking and other types of thinking skills. They compared computational thinking with mathematical, engineering, design and system thinking. In their article, they described the relationship among computational thinking, computer science and programming [20].

The application of mathematics is required for mathematical thinking to solve mathematical problems, for example, equations and functions [21]. Both computational thinking and mathematical thinking are essential for problem-solving processes [15].

In engineering, skills are necessary to create or develop things to better life [22]. Computational thinking and engineering together involve problem solving and the ability to understand the operation of complex systems [15].

The skill of understanding different relationships between elements in a predefined environment is included in system thinking [23]. Wing said that people who own the skill of system thinking understand a problem more easily and can theoretically model the workflow of a system. These people are able to manipulate the system model using computational tools, and they can understand the behavior of the given model. The ability to understand and model systems is necessary for computational and system thinking. Computational thinking involves more skills than system thinking. The latter mainly deals with identifying and understanding the operation of a system.

The ability to think as a designer and solve problems with this skill is required in design thinking [24]. Design thinking, as well as computational thinking, is a problem-solving process. Design thinking, for example, engineering, deals with product specifications and requirements that are also affected by humans and the environment. In contrast, computational thinking does not have physical limits to solve theoretical and practical problems.

The ability to program is one of the benefits of being able to think computationally, but programming skills are not matched with computational thinking skills [25,26]. The topic of computer science contains more than just learning about programming, and computational thinking involves more knowledge and skills than computer science [15]. Computational thinking deals with thinking about activities and problems in daily life.

The above articles provided an excellent basis for our article, defining computational thinking in general, as well as its cornerstones and dimensions. In our view, students' computational thinking cannot be developed in all areas, because ICT is a very broad concept. We focus more on the narrowed area of what we define as particular computational thinking. So, particular computational thinking is a narrowed subfield of computational thinking; in our case, we focus on the OO programming subfield. Based on the articles, we can conclude that the cornerstones of particular computational thinking are abstractions and algorithms, which are also their dimensions. In the below, we present some scientific findings that support our experience and view whereby the use of robots and microcontrollers is effective in programming education. They help to develop particular computational thinking.

One way to develop computational thinking is to deal with computer programming and robotics. Computational thinking is a term that has many definitions and includes skills such as analytic and problem-solving skills and other terms such as habits, dispositions and approaches related to computer science [27].

It was found from Bers, Flannery, Kazak and Sullivan's research study that four-year-old children could learn many concepts by playing and engaging with construction-based games based on robotics construction activities. TangibleK Robotics Program, where they focus on developing children's skills such as robotics, problem solving, programming and computational thinking in kindergarten, is based on a constructionist curriculum where they connect robotics tools with programming abilities, and it is designed for the appropriate age groups. The project reveals the strengths and weaknesses of the curriculum. The main result of this project was that kindergartens were enthusiastic and could acquire new skills about computational thinking, programming and robotics [1]. This project was introduced in an article, which explained that young children can actively engage in learning computer programming with the help of robotic tools that they use to develop their computational thinking skills. This statement depends on age-appropriate technologies, curricula and pedagogies.

In Christina Chalmers' article, it is shown how teachers integrated programming and robotics into classrooms activities and how it changed students' computational thinking as a result of an Australian research study [28]. During this research study, teachers introduced the LEGO and WeDO 2.0 robotics kits to their students. She declared that these methods developed computational thinking and suggested the use of these robotics kits in primary schools to develop problem-solving skills [28–30].

The authors of the article [31] presented the result of a research study in which they observed the development of secondary school students' computational thinking skills using computing instructions. The research study was based on ten tasks selected from Bebras challenges. It was proven that there were no significant differences between students' computational thinking results in terms of gender.

Their results were also confirmed in studies by two authors: Kalas and Tomcsányiová. According to them, there are no significant differences in gender-based performances, which was proved by Bebras test scores [7,32].

Professional visualization is essential for the development of computational thinking during programming. Students can better master the basic concepts and control structures

related to programming with the help of visualization. They gain a deeper understanding [33]. Robots and microcontrollers are very good tools for visualization. There are many more options for secondary school programming, as students' intellectual maturity allows us to also take advantage of more sophisticated microcontrollers. Lego Mindstorms robot programming is very common. In a graphical development environment, students can easily master control structures and the operation and use of sensors. Lego Mindstorms also allows one to use other programming languages, such as Python, C and Java.

The potential of Lego Mindstorms in providing an opportunity to test the knowledge of talented students is being recognized by more and more countries. There are races where a special robot has to be built and then programmed. Such competition categories are: robot sumo, line tracking, maze and weightlifting, completing different types of obstacle courses, throwing and firefighting.

Sergei A. Filippov and his colleagues have a few years of experience in teaching robotics at secondary schools and showed that the interest in LEGO robot structures decreased in older students (15–17 years). The reason for this was that they were oriented towards the university entrance exam and focused on other subjects instead of robotics. Younger students (10–14 years old) showed greater interest in robotics as an interesting complex game. The research study showed a challenge, how to motivate older students to engage with robotics, programming and mathematics. A laboratory was set up in which high-school students worked with university students on five projects. At the end of the projects, they came to the conclusion that LEGO Mindstorms met the main educational and vocational orientation needs of high-school students, with particular reference to automation, control systems, robotics, etc. [34]. The mutual interest and understanding in the collaboration between university students and high-school students were important and fascinating [34–36].

More and more secondary schools are working to program the microcontrollers of the Arduino family and Raspberry Pi [37,38]. It is necessary to have a basic knowledge of electricity and electronics. There are projects where no basic knowledge is required. Knowledge is acquired during the project. One such well-known project is Digital Magic, where a basic electronics package is provided, and smaller projects are built on this. Tasks that engage and arouse interest are provided.

Technical secondary schools are already immersed in the world of microcontrollers. During their studies, they become familiar with electronic components and the structure of processors. In addition to programming the former robot and microcontroller types, they are also familiar with other special microcontrollers programmed by using lower-level programming languages, such as the PIC microcontroller in the Assembly programming language.

Dogan Ibrahim made the following suggestions for teaching microcontrollers:

- Methods of architecture and assembly instruction are nowadays outdated topics to be taught; they have been replaced by methods of teaching the microcontroller;
- For teaching the architectures of famous microcontrollers, one needs few minutes to understand;
- Teaching students to use a high-level language (such as C) to program the microcontroller: There is little time to teach the Assembly language, as, today, most microcontrollers are based on the usage of a high-level programming language, which makes programming and maintenance easier;
- Using the development kit, we can configure the board to work with different microcontroller chips. The article teaches the students to use a variety of microcontrollers. Moving a program should be an easy task, as a program developed in high-level programming language can be easily moved from one platform to another, because this type of language usually is independent from the type of the microcontroller [39].

The eight dimensions of the averaged computational thinking skills of Anita Juškevičienė can be best illustrated in object-oriented planning and programming in programming education, as all dimensions appear during development. In the course of our

research study, the research results of the last 5 years of the OO planning and programming subject of Budapest Business School were processed.

During the investigation, 3296 test results were examined using IBM SPSS Statistics software. Table 1 shows the proportion of women, 29.1% (958 people), and the proportion of men, 70.9% (2338 people).

Table 1. Gender distribution.

| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|-------|--------|-----------|---------|---------------|--------------------|
| Valid | Female | 958 | 29.1 | 29.1 | 29.1 |
| | Male | 2338 | 70.9 | 70.9 | 100.0 |
| | Total | 3296 | 100.0 | 100.0 | |

Based on the distribution obtained from the filtered data, it can be stated that the given OOP subject was taken by many more male students, almost two and a half times more, than female students. This is due to the fact that IT subjects in a business school are generally more attractive to men.

The study group consisted of full-time and correspondence students; see Tables 2 and 3.

Table 2. Training-type distribution.

| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|-------|------------------------|-----------|---------|---------------|--------------------|
| Valid | Corresponding training | 709 | 21.5 | 21.5 | 21.5 |
| | Full-time training | 2587 | 78.5 | 78.5 | 100.0 |
| | Total | 3296 | 100.0 | 100.0 | |

Table 3. Distribution of training types by gender.

| | | Total | | Male | | Female | |
|-------|------------------------|-----------|---------|-----------|---------|-----------|---------|
| | | Frequency | Percent | Frequency | Percent | Frequency | Percent |
| Valid | Corresponding training | 709 | 21.5 | 481 | 67.8 | 228 | 32.2 |
| | Full-time training | 2587 | 78.5 | 1857 | 71.8 | 730 | 28.2 |
| | Total | 3296 | 100.0 | 2338 | | 958 | |

The data in the table show that there were more than three-and-a-half times more full-time students, which shows the proportion of headcount allowed to be achieved by the university between the two training methods.

Table 3 shows that in our sample, the proportion of women in correspondence training was higher—nearly 1/3—while in the case of full-time training, the number of male students was 2.5 times higher than that of women.

The average score for all students was 2.18, where 1 was the weakest (failed exam) and 5 the best (Table 4). This average reflects the fact that the students found it very difficult to pass the exam. The number of test applications shows how many times the exam was taken. It can be seen that the range was large (5), that is, there were students who tried (repeated the topic) six times. Most of them passed the exam the first time, but the average did it the second time.

Table 5 shows how many students passed the exam but does not clearly show how many of them were successful. The following table (Table 6) summarizes the number of successful exams from time to time. A non-1 exam result counts as a successful exam. The first two rows of the table show that the number of students who successfully passed the second exam was much higher than the number of students who failed the first exam. The reason behind this was the dissatisfaction of many students with their exam result, so they repeated it, hoping for a better result. If we examine the percentage of successful exams in

relation to the possibilities of the exam, we can see that success decreases linearly. It was actually the expected result, because it is harder to pass the exam for less talented students.

Table 4. Statistics on the number of exam applications and grade of all students.

| | | Grade | Number of Applications for Exam |
|--------------------|---------|-------|---------------------------------|
| N | Valid | 3296 | 3296 |
| | Missing | 0 | 0 |
| Mean | | 2.18 | 1.97 |
| Std. Error of Mean | | 0.022 | 0.019 |
| Median | | 2.00 | 2.00 |
| Mode | | 1 | 1 |
| Std. Deviation | | 1.239 | 1.089 |
| Variance | | 1.536 | 1.185 |
| Range | | 4 | 5 |
| Minimum | | 1 | 1 |
| Maximum | | 5 | 6 |

Table 5. Number of applications for exam distribution of all students.

| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|-------|-------|-----------|---------|---------------|--------------------|
| Valid | 1 | 1386 | 42.1 | 42.1 | 42.1 |
| | 2 | 1095 | 33.2 | 33.2 | 75.3 |
| | 3 | 477 | 14.5 | 14.5 | 89.7 |
| | 4 | 216 | 6.6 | 6.6 | 96.3 |
| | 5 | 110 | 3.3 | 3.3 | 99.6 |
| | 6 | 12 | 0.4 | 0.4 | 100.0 |
| | Total | 3296 | 100.0 | 100.0 | |

Table 6. Grade distribution by number of applications of all students.

| | | Total | | Successful Exams | | Failed Exams | |
|-------|-------|-----------|---------|------------------|---------|--------------|---------|
| | | Frequency | Percent | Frequency | Percent | Frequency | Percent |
| Valid | 1 | 1386 | 42.1 | 1250 | 90.2 | 136 | 9.8 |
| | 2 | 1095 | 33.2 | 558 | 51.0 | 537 | 49.0 |
| | 3 | 477 | 14.5 | 146 | 30.6 | 331 | 69.4 |
| | 4 | 216 | 6.6 | 53 | 24.5 | 163 | 75.5 |
| | 5 | 110 | 3.3 | 23 | 20.9 | 87 | 79.1 |
| | 6 | 12 | 0.4 | 2 | 16.7 | 10 | 83.3 |
| | Total | 3296 | 100.0 | 2032 | | 1264 | |

The table shows that the more times one tried to take the exam, the less likely the exam was successful. The reason was that the students were very prepared for the first exam, and they took contact classes seriously, while in the case of the latter, the material taught in the contact lesson and the knowledge learned were farther and farther away in the past, so the student were less prepared for the exam.

The skewed distribution in Figure 1 shows the number of exams taken according to Table 3.

In the examination system, we used a 5-point grading scale, with the worst being mark 1 and the best being mark 5. In both cases, we obtained a decreasing distribution towards a higher value, which is to be expected, since for some students, only success matters, not what results from success. The distribution for the entire sample is shown in Figure 1.

Table 7 shows the percentage of failed exams, 38.3%. The least successful was 29.7%, and the number of better exam results was almost halved. Only 7% of the exams ended with the best results.

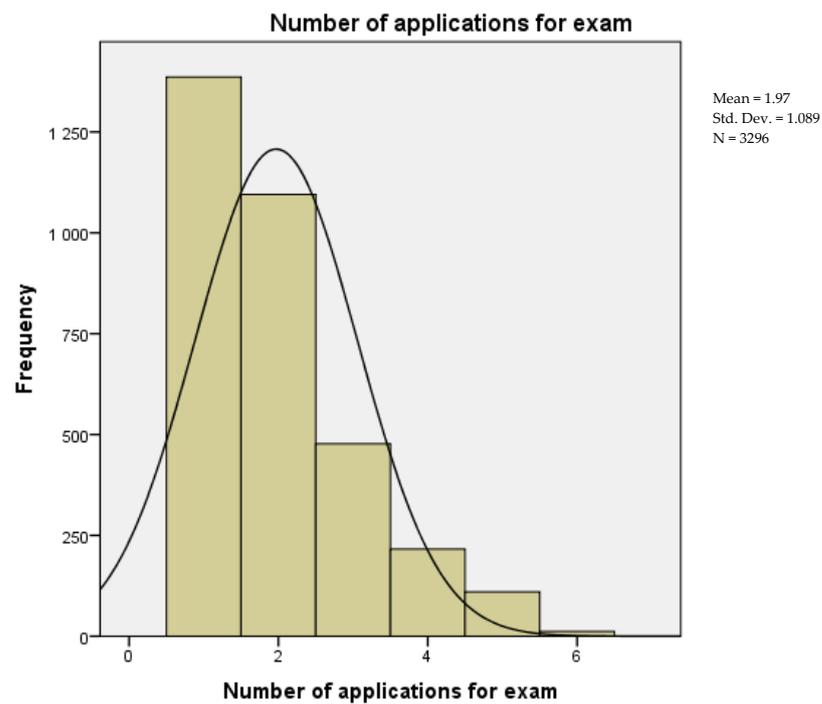


Figure 1. Histogram of the number of applications for the exam of all students.

Table 7. Grade distribution by exam.

| | Frequency | Percent | Valid Percent | Cumulative Percent |
|-------|-----------|---------|---------------|--------------------|
| Valid | 1 | 1264 | 38.3 | 38.3 |
| | 2 | 980 | 29.7 | 68.1 |
| | 3 | 488 | 14.8 | 82.9 |
| | 4 | 333 | 10.1 | 93.0 |
| | 5 | 231 | 7.0 | 100.0 |
| Total | 3296 | 100.0 | 100.0 | |

When examining the exam results per semester (Table 8), the number of exams was as shown in the table below (OO planning and programming were not taught in the semester of 2019/20/1).

Table 8. Distribution of numbers of exams.

| | Frequency | Percent | Valid Percent | Cumulative Percent |
|-------|-----------|---------|---------------|--------------------|
| Valid | 2015/16/1 | 334 | 10.1 | 10.1 |
| | 2015/16/2 | 366 | 11.1 | 21.2 |
| | 2016/17/1 | 295 | 9.0 | 30.2 |
| | 2016/17/2 | 535 | 16.2 | 46.4 |
| | 2017/18/1 | 337 | 10.2 | 56.6 |
| | 2017/18/2 | 559 | 17.0 | 73.6 |
| | 2018/19/1 | 150 | 4.6 | 78.2 |
| | 2018/19/2 | 410 | 12.4 | 90.6 |
| | 2019/20/2 | 310 | 9.4 | 100.0 |
| Total | 3296 | 100.0 | 100.0 | |

Considering these data, it is worth noting that the subject was delivered during the contact hour in the second semester. In the 1st semester, the student could take the subject with individual preparation if he/she had already taken it in the previous semester as a contact hour. Thus, in the first semester, there could only be students who had failed the

subject in the previous semester, but there were still many unsuccessful completions, as there was a 43% reduction in the number of students in this semester.

Table 9 shows the success per semester. The second semester showed much greater success.

Table 9. Successfulness distribution by semester.

| | Total | | Successful Exams | | Failed Exams | | |
|-------|-----------|---------|------------------|---------|--------------|---------|------|
| | Frequency | Percent | Frequency | Percent | Frequency | Percent | |
| Valid | 2015/16/1 | 334 | 10.1 | 227 | 68.0 | 107 | 32.0 |
| | 2015/16/2 | 366 | 11.1 | 226 | 61.7 | 140 | 38.3 |
| | 2016/17/1 | 295 | 9.0 | 212 | 71.9 | 82 | 28.1 |
| | 2016/17/2 | 535 | 16.2 | 244 | 45.6 | 291 | 54.4 |
| | 2017/18/1 | 337 | 10.2 | 237 | 70.3 | 100 | 29.7 |
| | 2017/18/2 | 559 | 17.0 | 228 | 40.8 | 331 | 59.2 |
| | 2018/19/1 | 150 | 4.6 | 136 | 90.7 | 14 | 9.3 |
| | 2018/19/2 | 410 | 12.4 | 227 | 55.4 | 183 | 44.6 |
| | 2019/20/2 | 310 | 9.4 | 294 | 94.8 | 16 | 5.2 |
| | Total | 3296 | 100.0 | 2032 | | 1264 | |

Figure 2 does not show a continuous increase in efficiency, i.e., decrease. The values varied by semester. However, the trendline showed a slight increase. On the basis of these, we can state that the exam results of the nine semesters did not show a significant improvement but rather stagnation, although, based on our digital environment and the interest in digital devices, we could assume that student performance improved. To improve this, new methods would need to be applied in education.

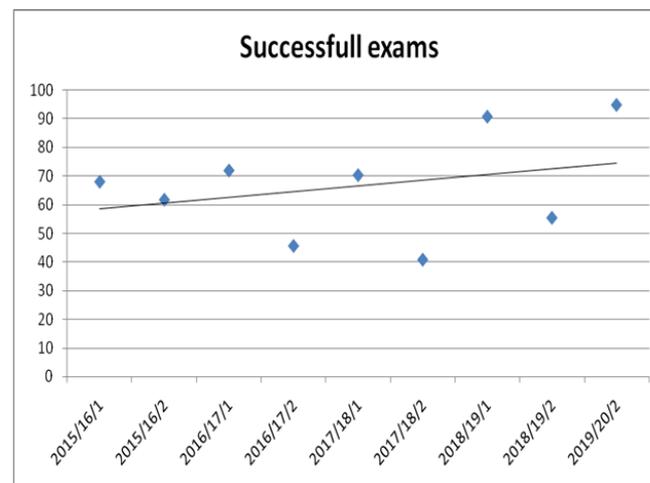


Figure 2. Trend of success by semester of all students.

4. Educational Hypotheses

In the course of our research study, we could define three hypotheses based on data processing.

H1. *The level of particular computational thinking can be measured using the exam results.*

Our experience shows that in the case of minimal completion of the exam, students did not develop particular computational thinking during the given contact hour and individual preparation. This can be justified by the fact that in this case, the students only tried to pass the exam, i.e., to complete the subject, and not to develop particular computational thinking. This also means that in the case of students striving for a minimum completion of the subject of OO programming, they strived to develop other particular

computational thinking (ERP knowledge, process control, database management, etc.). In all cases, students who achieved better results than the minimum results showed more work and the development of particular computational thinking. The results are also supported by Figure 3, where the histogram spectacularly shows the results achieved by the students. It can be seen that the students only strived for the minimum completion of the subject. The reason for this may also be that the direction of the university is not explicitly programming or software development training but also business IT training.

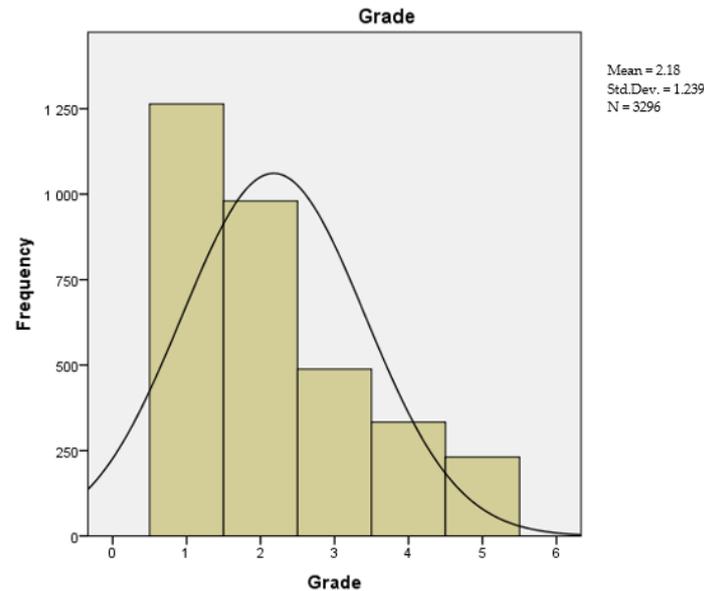


Figure 3. Histogram of the grade distribution by exam of all students.

H2. *Teacher advancement (experience) improves the level of particular computational thinking.*

Figure 2 supports our hypothesis. We can observe that in the case of student success, the trendline showed a slight increase, which also brings with it the development of particular computational thinking. Compared with the beginning, our instructors took part in several trainings, where they learned new teaching methods and trends. In recent years, they gained internal experience in education. This is an unexpected result, as we could have had the opposite result based on the tiredness, burnout and routine teaching of the instructors.

H3. *The production of particular computational thinking during COVID-19 or online is much more effective than the pursuit of full computational thinking through traditional education.*

The reason for this is as follows: In the context of online education, the student is left much more to his/her own research work and his/her own search; therefore, with targeted guidance, it is much easier to achieve a narrower range of knowledge. They do not have to cover such a wide area, and it is easier to control what they review online. It is much easier for teachers to control the knowledge they acquire. Small gaps are also easier to spot than a major gap in a larger area. This finding was the result of COVID-19 and led to the realization that it is indeed the development of particular computational thinking (knowledge) that should be pursued. On the other hand, this is a much better way to separate the individual knowledge of the students and to build on individual knowledge than group activities.

5. Discussion

Based on the results of the Slovak secondary-school pilot project, we recommend that microcontroller programming should also be included in university education [37]. Here, we mean non-technically oriented universities; there is a separate subject on the

programming of microcontrollers (e.g., robotics). Students can learn about the technical-practical applications of object-oriented programming and thus develop their particular computational thinking.

Then, we can ask an important question: during the semester, how many opportunities should be devoted to teaching this topic within the object-oriented planning and programming subject? There are 12-14 lectures per semester, and the same number of practical lessons are available, occasionally 90 min/lesson.

We formulate three statements for which evidence needs to be obtained after the topic is introduced into education. By then, we should have enough information to prove our statements.

S1. *For the development of particular computational thinking within the subject of object-oriented planning and programming, which should be taught via the programming of microcontrollers, once or twice is not enough to achieve significant improvement (contact hours and at least the same number of individual hours to practice the curriculum).*

Interpreted in this way, in our opinion, 1-2 occasions are sufficient for students to become familiar with the technical design, applicability and performance of microcontrollers and to gain theoretical knowledge of the operating principles of electronic components, as well as to control structures. It should be noted that most students have secondary school education in which they have not encountered electronics.

S2. *For the development of particular computational thinking within the subject of object-oriented planning and programming, the programming of microcontrollers should be taught three times (and additional individual occupations), which are optimal to achieve significant improvement.*

In the third lesson, the students would gain real knowledge of microcontroller programming, followed by the development of particular computational thinking using microcontrollers. The microcontroller is considered a new “thing” because few have dealt with it before, although they can be found almost anywhere in real life. Therefore, the students show great interest in their topic. This is when students’ creativity can best be developed according to the eight dimensions of computational thinking defined by Juškevičienė [11,12].

S3. *The development of particular computer thinking within the subject of object-oriented design and programming, which should be taught through the programming of microcontrollers, does not result in a significant improvement even if they have more than 3 classes.*

We define our statements more than three times based on educational experience in secondary schools. Experience shows that over time, microcontrollers, such as robots, appear to students as intelligent games. Their application develops particular computational thinking but not anymore to the extent it would be expected.

Our research study was conducted just before the COVID-19 boom. It would also be interesting to examine, evaluate and compare the results of the students at the institution during the pandemic period. This can be the basis for a future study. In the below, we examine the relationship of our research study to the COVID-19 pandemic. The effects of COVID-19 on society (lockdowns, social distancing, closure of certain facilities, etc.), perceptions of risk and negative externalities have been addressed in several articles. The epidemic has also had a major impact on education.

The COVID-19 virus has had a major impact on the lives of many people worldwide. In Switzerland, the authors of [40] conducted a study in the German-speaking part of the country to assess the impact of trust and risk perception on the adoption of measures to reduce COVID-19 cases. The results suggest that the measures implemented were accepted by the public. Survey respondents were more concerned about other family members catching the virus than themselves and were also concerned about the economic impact. The results suggest that how trust is measured is key, as general trust and social trust have opposite effects on participants’ perceptions of risk. This result is also relevant for us. Trust

is also important for online education. In the case of measuring the level of individual knowledge online, it is important that students do not abuse the trust of the instructors by using impermissible instruments in the assessment. This is indeed a risk in judging the veracity of the results. The two research results presented below show that the best defense during a pandemic is social distancing.

Social distancing is one of the most recommended policies worldwide to reduce the risk of contact during the COVID-19 pandemic. From a risk-management perspective, the authors of [41] explored the mechanism of the risk aversion effect on social distancing to improve individual physical distancing behavior. Their results show that risk perception had a significant positive effect on perceived understanding and social distancing. Perceived understanding showed a significant positive correlation with social distancing behaviors and played a mediating role in the relationship between risk perception and social distancing behaviors.

Social distancing has played a critical role in reducing the risk of disease transmission during the COVID-19 pandemic and post-epidemic period. To explore social distancing obedience behavior, the authors of [42] conducted a comprehensive survey, collecting data from Chinese residents using a questionnaire. The results of the conducted analyses show that public guidance significantly influenced individuals' risk perception, while risk perception had a positive effect on social distancing obedience behavior. Moreover, risk perception plays a mediating role in the relationship between public guidance and social distancing compliance behavior. In addition, regulatory punishment positively predicts social distancing obedience behavior and can have an even greater effect by increasing risk-taking. Social distancing can make traditional contact-hour education unfeasible, so all countries have reacted quickly by enabling and introducing online education.

The authors in [43] evaluated learning engagement, satisfaction and anxiety levels using an e-book-based distance-learning strategy on an online learning platform. In a Japanese urban high school, learning logs were analyzed to measure students' engagement, while survey responses indicated their perceptions of the distance-learning experience. The diary analysis showed that the average completion rate for 267 subjects was 67%. A significant decrease in engagement was also observed 3 weeks after distance learning and across subjects and grades. The analysis of the survey showed that students also felt satisfaction and anxiety about distance learning. However, there were significant differences in the level of satisfaction among the different grades. The results showed that maintaining student motivation is a challenge in secondary distance education and that we need to alleviate students' concerns about their own progress after the break.

In [44], the authors examined distance-education students' perspectives on online learning based on their accessibility modules (accessibility and functionality), online presentation or teaching methods, actual use of online learning and challenges of online learning. The study found statistically significant relationships between the accessibility and functionality of online widgets, the accessibility of online modules and online presentation methods, functionality and online presentation methods, and functionality and online learning use. Thus, these factors are necessary for successful online learning. The four main perceived challenges were unreliable power/electricity and internet connection, lack of collaboration and lack of motivation during online learning. Therefore, it was suggested that steps should be taken to address the perceived challenges while using a blended approach involving face-to-face and online learning. The blended learning approach would allow students to better adapt and appreciate the usefulness of distance learning.

In another article, a survey was carried out with students studying health-related courses at a university on how their lifestyle, behavior, mental health and education were affected by the pandemic. In general, it had a negative impact on behavior, lifestyle and mental health, and virtual education was seen as necessary to make up for the loss of personal experience. The current pandemic has affected students' mental health and educational needs, and health education institutions must respond to these needs to ensure that students continue to receive the support they need [45].

No very clear conclusions can be drawn from the results of the articles reviewed. The results are strongly influenced by the general national, environmental and school-type conventions that have been established. In secondary education, most studies show deteriorating results. The reasons for this should certainly be investigated. For students and teachers, online education is a new situation. The trend when COVID-19 was introduced was to take over all the curriculum, which is not possible in the new environment. Later on, most countries realized that this was not the best solution and reduced the curriculum. Education is fully supported by computing. This proves that it is not possible to develop the whole computational thinking at once, but only in a particular way. The situation is different in university education when teaching online. Students are not far from individual preparation, but here too, it is a question of developing individual computational thinking.

6. Conclusions

In the course of our research study, we examined the exam results of the last 5 years of the object-oriented planning and programming subject of the Budapest Business School. The results show that the success of the exams improved minimally per semester. However, the examination of the grades revealed that the average of the exam results was barely above the success threshold (the average was 2.18, and the rate of failed exams was 38.3%). Particular computational thinking must be developed to achieve better results. In researching the literature, several methods were found with which the authors had positive experiences in the teaching of university object-oriented planning and programming. In our opinion, the introduction of microcontroller programming would be an appropriate way to develop particular computational thinking. The four important key techniques (cornerstones) of computational thinking defined by Wing, decomposition, pattern recognition, abstraction and algorithms, can be applied well in the teaching of programming using microcontrollers. Students can gain deeper knowledge and experience with the help of microcontrollers that can be used in real life. We found that 3 of the 12-14 lessons available during the semester should be devoted to programming microcontrollers. Such a number of lessons is considered to be optimal for students to gain practical experience in the field of microcontroller programming and to develop particular computational thinking as optimally as possible with the help of microcontrollers. Based on secondary school educational practice, fewer options are considered, while more are considered unnecessary because more lessons no longer contribute significantly to the development of particular computational thinking.

Furthermore, we proved our hypotheses, which are as follows: “The level of particular computational thinking can be measured from the exam results.” and “Teacher advancement (experience) improves the level of particular computational thinking.”. Particular computing thinking does not depend on the number of hours but primarily on commitment. “The production of particular computational thinking during COVID-19 or online is much more effective than the pursuit of full computational thinking through traditional education.”

We consider it important in online education that the individual tools used by students can be integrated into education, without the need to purchase additional tools or to make them more complex. These operations are very important from a sustainability point of view, as well as from a recycling point of view. Because of the increase in energy consumption and device utilization, the sustainability is much higher, so the pursuit of online individual knowledge always results in lower emissions.

Future research could investigate how the whole field of computational thinking (knowledge) can be clustered into particular ways of thinking (knowledge). What relationships can be identified among these particular domains? How can parameters be defined so that the total computational knowledge is captured with the least overlap? If this is the case, the other areas could be examined from the same perspective.

Author Contributions: Conceptualization, Á.G., J.U. and M.T.F.; methodology, Á.G., J.U. and Á.S. writing—original draft preparation, Á.G., J.U., M.T.F. and Á.S.; writing—review and editing, J.U. and M.T.F.; visualization, J.U. and Á.S.; supervision, M.T.F.; project administration, M.T.F., J.U. and Á.S.; funding acquisition, J.U., Á.G. and Á.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research project was funded by Budapest Business School Research Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This research project was carried out within the framework of Centre of Excellence for Future Value Chains of Budapest Business School.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Bers, M.U.; Flannery, L.; Kazakoff, E.R.; Sullivan, A. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Comput. Educ.* **2014**, *72*, 145–157. [\[CrossRef\]](#)
- Chiassese, G.; Arrigo, M.; Chifari, A.; Lonati, V.; Tosto, C. Educational robotics in primary school: Measuring the development of computational thinking skills with the bebras tasks. *Informatics* **2019**, *6*, 43. [\[CrossRef\]](#)
- Piedade, J.; Dorotea, N.; Pedro, A.; Matos, J.F. On teaching programming fundamentals and computational thinking with educational robotics: A didactic experience with pre-service teachers. *Educ. Sci.* **2020**, *10*, 214. [\[CrossRef\]](#)
- Avornicului, M.C.; Seer, L.; Benedek, B. Identitás a XXI. sz. információs társadalmában, az internet hatásai. *Logiszt. Inform. Menedzsment* **2016**, *2016*, 70–74.
- Llorens-Largo, F. Dicen por ahí . . . que la nueva alfabetización pasa por la programación. *ReVisión* **2015**, *8*, 11–14.
- Brown, N.C.C.; Kolling, M.; Crick, T.; Peyton Jones, S.; Humphreys, S.; Sentance, S. Bringing computer science back into schools: Lessons from the UK. Proceeding of the 44th ACM Technical Symposium on Computer Science Education, SIGCSE '13, Denver, CO, USA, 6–9 March 2013; ACM: New York, NY, USA, 2013; pp. 269–274.
- Acevedo-Borrega, J.; Valverde-Berrosco, J.; Garrido-Arroyo, M.D.C. Computational Thinking and Educational Technology: A Scoping Review of the Literature. *Educ. Sci.* **2022**, *12*, 39. [\[CrossRef\]](#)
- García-Penalvo, F.J.; Llorens Largo, F.; Molero Prieto, X.; Vendrell Vidal, E. Educación en Informática sub 18 (EI <18). *ReVision* **2017**, *10*, 13–18.
- Akram, U.; Fülöp, M.T.; Tiron-Tudor, A.; Topor, D.I.; Căpușneanu, S. Impact of digitalization on customers' well-being in the pandemic period: Challenges and opportunities for the retail industry. *Int. J. Environ. Res. Public Health* **2021**, *18*, 7533. [\[CrossRef\]](#)
- Weintrop, D.; Beheshti, E.; Horn, M.; Orton, K.; Jona, K.; Trouille, L.; Wilensky, U. Defining computational thinking for mathematics and science classrooms. *J. Sci. Educ. Technol.* **2016**, *25*, 127–147. [\[CrossRef\]](#)
- Juškevičienė, A. STEAM Teacher for a Day: A Case Study of Teachers' Perspectives on Computational Thinking. *Inform. Educ.* **2020**, *19*, 33–50. [\[CrossRef\]](#)
- Juškevičienė, A. Developing Algorithmic Thinking Through Computational Making. In *Data Science: New Issues, Challenges and Applications*; Springer: Cham, Switzerland, 2020; pp. 183–197.
- Kothari, C.R.; Garg, G. *Research Methodology—Methods and Techniques*, 4th ed.; New Age International (P), Limited Publishers: New Delhi, India, 2019.
- Davis, C. *SPSS for Applied Sciences: Basic Statistical Testing*; CSIRO Publishing: Collingwood, Australia, 2013.
- Wing, J.M. Computational thinking. *Commun. ACM* **2006**, *49*, 33–35. [\[CrossRef\]](#)
- Berland, M.; Wilensky, U. Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *J. Sci. Educ. Technol.* **2015**, *24*, 628–647. [\[CrossRef\]](#)
- Wing, J.M. Computational thinking. In Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing, Pittsburgh, PA, USA, 18–22 September 2011; p. 3.
- Brennan, K.; Resnick, M. New frameworks for studying and assessing the development of computational thinking. Presented at the Annual American Educational Research Association Meeting, Vancouver, BC, Canada, 13–17 April 2012.
- Juškevičienė, A.; Dagiene, V. Computational Thinking Relationship with Digital Competence. *Inform. Educ.* **2018**, *17*, 265–284. [\[CrossRef\]](#)
- Shute, V.J.; Sun, C.; Asbell-Clarke, J. Demystifying computational thinking. *Educ. Res. Rev.* **2017**, *22*, 142–158. [\[CrossRef\]](#)
- Sneider, C.; Stephenson, C.; Schafer, B.; Flick, L. Computational thinking in high school science classrooms. *Sci. Teach.* **2014**, *81*, 53–59. [\[CrossRef\]](#)
- Bagiati, A.; Evangelou, D. Practicing engineering while building with blocks: Identifying engineering thinking. *Eur. Early Child. Educ. Res. J.* **2016**, *24*, 67–85. [\[CrossRef\]](#)
- Shute, V.J.; Masduki, I.; Donmez, O. Conceptual framework for modeling, assessing, and supporting competencies within game environments. *Technol. Instr. Cogn. Learn.* **2010**, *8*, 137–161.

24. Razzouk, R.; Shute, V. What is design thinking and why is it important? *Rev. Educ. Res.* **2012**, *82*, 330–348. [[CrossRef](#)]
25. Ioannidou, A.; Bennett, V.; Reppenning, A.; Koh, K.H.; Basawapatna, A. Computational thinking patterns. Presented at the Annual Meeting of the American Educational Research Association, New Orleans, LA, USA, 8–12 April 2011.
26. Ionescu, C.A.; Fülöp, M.T.; Topor, D.I.; Căpușeanu, S.; Breaz, T.O.; Stănescu, S.G.; Coman, M.D. The New Era of Business Digitization through the Implementation of 5G Technology in Romania. *Sustainability* **2021**, *13*, 13401. [[CrossRef](#)]
27. Barr, V.; Stephenson, C. Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads* **2011**, *2*, 48–54. [[CrossRef](#)]
28. Chalmers, C. International Journal of Child-Computer Interaction Robotics and computational thinking in primary school. *Int. J. Child-Comput. Interact.* **2018**, *17*, 93–100. [[CrossRef](#)]
29. Balcombe, L.; De Leo, D. Human-Computer Interaction in Digital Mental Health. *Informatics* **2022**, *9*, 14. [[CrossRef](#)]
30. Sanchez, C.; Costa, V.; Garcia-Carmona, R.; Urendes, E.; Tejedor, J.; Raya, R. Evaluation of Child-Computer Interaction Using Fitts' Law: A Comparison between a Standard Computer Mouse and a Head Mouse. *Sensors* **2021**, *21*, 3826. [[CrossRef](#)] [[PubMed](#)]
31. Delal, H.; Oner, D. Developing Middle School Students' Computational Thinking Skills Using Unplugged Computing Activities. *Inform. Educ.* **2020**, *19*, 1–13. [[CrossRef](#)]
32. Kalaš, I.; Tomcsányiová, M. Students' attitude to programming in modern informatics. In Proceedings of the 9th World Conference on Computers in Education, Bento Gonçalves, Brazil, 27–31 July 2009; p. 82.
33. Udvaros, J.; Gubán, M. Demonstration the class, objects and inheritance concepts by software. *Didact. Napoc.* **2016**, *9*, 23–34.
34. Filippov, S.A.; Fradkov, A.L.; Andrievsky, B. Teaching of robotics and control jointly in the University and the high school based on LEGO Mindstorms NXT. In Proceedings of the 18th IFAC World Congress on Automatic Control, Milan, Italy, 28 August–2 September 2011; pp. 9824–9829. [[CrossRef](#)]
35. Cano, S. A Methodological Approach to the Teaching STEM Skills in Latin America through Educational Robotics for School Teachers. *Electronics* **2022**, *11*, 395. [[CrossRef](#)]
36. Pozzi, M.; Prattichizzo, D.; Malvezzi, M. Accessible educational resources for teaching and learning robotics. *Robotics* **2021**, *10*, 38. [[CrossRef](#)]
37. Udvaros, J.; Végh, L. New teaching methods by using microcontrollers in teaching programming. *Elearning Softw. Educ.* **2020**, *1*, 630–637. [[CrossRef](#)]
38. Jacko, P.; Bereš, M.; Kováčová, I.; Molnár, J.; Vince, T.; Dziak, J.; Fecko, B.; Gans, Š.; Kováč, D. Remote IoT Education Laboratory for Microcontrollers Based on the STM32 Chips. *Sensors* **2022**, *22*, 1440. [[CrossRef](#)]
39. Ibrahim, D. A new approach for teaching microcontroller courses to undergraduate students. *Procedia-Soc. Behav. Sci.* **2014**, *131*, 411–414. [[CrossRef](#)]
40. Siegrist, M.; Luchsinger, L.; Bearth, A. The Impact of trust and risk perception on the acceptance of measures to reduce COVID-19 cases. *Risk Anal.* **2021**, *41*, 787–800. [[CrossRef](#)] [[PubMed](#)]
41. Xie, K.; Liang, B.; Dulebenets, M.A.; Mei, Y. The impact of risk perception on social distancing during the COVID-19 pandemic in China. *Int. J. Environ. Res. Public Health* **2020**, *17*, 6256. [[CrossRef](#)] [[PubMed](#)]
42. Yuan, J.; Zou, H.; Xie, K.; Dulebenets, M.A. An Assessment of Social Distancing Obedience Behavior during the COVID-19 Post-Epidemic Period in China: A Cross-Sectional Survey. *Sustainability* **2021**, *13*, 8091. [[CrossRef](#)]
43. Kuromiya, H.; Majumdar, R.; Miyabe, G.; Ogata, H. E-book-based learning activity during COVID-19: Engagement behaviors and perceptions of Japanese junior-high school students. *Res. Pract. Technol. Enhanc. Learn.* **2022**, *17*, 12. [[CrossRef](#)]
44. Segbenya, M.; Bervell, B.; Minadzi, V.M.; Somuah, B.A. Modelling the perspectives of distance education students towards online learning during COVID-19 pandemic. *Smart Learn. Environ.* **2022**, *9*, 13. [[CrossRef](#)]
45. Gadi, N.; Saleh, S.; Johnson, J.A.; Trinidad, A. The impact of the COVID-19 pandemic on the lifestyle and behaviours, mental health and education of students studying healthcare-related courses at a British university. *BMC Med. Educ.* **2022**, *22*, 115. [[CrossRef](#)]