edu{code}

TRAINING NEEDS ANALYSIS REPORT

Increasing The Competency of Computer Science Teaching Undergraduates On Coding Education

REPORTED BY APEC

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Project Information

Project name	Increasing The Competency of Computer Science Teaching Undergraduates On Coding Education
Project acronym	EDUCODE
Project number	2017-1-TR01-KA203-046578
IO1 – Lead partner	Dokuz Eylül University

Document Identification

Document title	Common Report – Training Need Analysis of Computer Science Teaching & STEM Undergraduates about Coding Education
Dissemination level	Project partners

Version number	Date released	Authors and contributors
1.00	March 05, 2018	• Dokuz Eylül University Kürşat Arslan Bahar Baran Ercan Akpınar • Limerick Institute of Technology Seamus Hoyne • Maribor University Igor Pesek Marjan Krasna Aleksander Vesel • Zagreb University Dragutin Kermek Predrag Oreški Goran Hajdin Danijel Radošević • APEC Yıldırım Özkaya Barış Özçelik

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

1. Executive Summary

This report is the first deliverable, prepared in the scope of The EU funded Erasmus+ Key Action 2 – The Strategic Partnership project entitled "Increasing The Competency of Computer Science Teaching Undergraduates on Coding Education - EDUCODE", outlining the training needs of computer science teaching & STEM undergraduates who will teach coding in primary and secondary schools.

It is a combination of the survey research work undertaken to identify the needs of future ICT & STEM teachers related to the delivery of coding education in mainstream schools.

The report features 7 chapters, each contributing to the aspect of identifying

1- The content of "Coding Education Training for Computer Science & STEM Undergraduates"
2- The work with respect to engagement with instructors (training providers), other beneficiaries and also STEM researchers
3- The grounds and principles of the evaluation of the training activities

Chapter 1 is the executive summary, providing information about the TNA document along with a brief explanation about what the training needs analysis addresses, and a summary of content for each chapter.

Chapter 2 provides an introduction, including an overview of the general research methodology carried out in the study together with brief explanations about coding education and its expected impacts on the future of education and further life experiences.

Chapter 3 discusses the survey methodology together with followed interactions, preparation processes, and compilation of questions, translations and application of the questionnaires on the online survey platforms.

Chapter 4 provides information regarding sampling in 4 partner universities, which include Dokuz Eylül University, Zagreb University, Limerick Institute of Technology and Maribor University. Statistics for the participants' personal information such as age, gender, year of study and the level of involvement in coding activities to date will be discussed in this section.

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Chapter 5 provides the process of deeper data analysis, mark ups which code text responses into numerical form, grouping data, checking the data for corrections, running tests to compare groups, and data visualization.

Chapter 6 identifies the input from the sampling and discusses the research findings in depth.

Chapter 7 provides a conclusion, which discusses the most prominent training needs revealed as a result of the analysis both in general and on the basis of each partner university sampling.

Training Needs Analysis

Training Needs Analysis addresses the following arguments.

- Identifying the gap between current and required levels of knowledge, skills and aptitude
- Identifying what the general content of training should be
- Forming the foundation of a training plan
- Providing a baseline for the evaluation of a training plan
- Ensuring that appropriate and relevant training is delivered

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Contents

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

2. Introduction

Technology revolutions, coming one after another are transforming our world. Digital tools are everywhere in today's society. We are encouraged to go online and communicate with others, store and share information, complete banking transactions, even sell, buy, order fast food or become engaged with various leisure activities online. The Internet has been leading the way in how we interact with our world for a number of decades.

Job definitions are also transforming for career advancement or change, many people need to learn new digital skills including developing new software and coding. Companies define themselves as “tech” companies even though they render services in far different areas such as catering, food and drink, or rent a car services. They deliver their products through the Internet and also open means of internal and external communication.

The European Commission also states, "each and every interaction between humans and computers is governed by code. Whether you create a web app, follow GPS directions when driving or wish to revolutionize social interactions. Coding is the literacy of today and it helps practice 21st century skills such as problem solving, team work and analytical thinking”.

Therefore, coding education is the future of both education and the development of ever emerging jobs, which require knowledgeable staff that know how to develop, update, or fix the problems of software.

Educators and experts also believe that there are other reasons to teach kids on how to code:

1- Learning to code prepares kids for the world we live in today.
   There are many jobs and occupations that use code directly, such as web designers, software developers and robotics engineers, and many more occupations where knowing how to use coding is a huge asset, such as jobs in manufacturing, nanotechnology or information sciences.
2- Understanding Code Helps Explain The World
   Today, computing is involved in almost all aspects of our lives, from communications and education to social media, banking, information, security and shopping. Networked computers are capable of controlling our homes’ thermostats and lighting, our cars and our health records.
3- Learning To Code Develops Problem Solving and Computational Thinking Skills

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

> The skills developed with computer programming help kids develop new ways of thinking and foster problem-solving techniques that can have positive repercussions in other areas of their lives.

Educode believe that the ability to contribute to software will improve employability, enhance creativity and raise the awareness of the cutting edge of innovations.

Thus the future teachers' awareness of the benefits of coding education is of special importance in terms of comprehending its necessity.

The partnership adopted itself to develop two essential intellectual outputs as follows:

1- Higher Education Curriculum to Increase Competency of Computer Science Teaching Undergraduates on Coding Education,
2- Lecturer Handbook to be used in parallel with the higher education curriculum

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

3. Survey Methodology

3.1 Initial Interactions

At the beginning of the process, members of the partner organizations corresponded through emails and scheduled the activities to prepare the Training Needs Analysis. A Skype meeting was held in order to decide and agree the content of the analysis.

It was agreed to consider and research the following questions:

- What to measure
- Which questions to ask / Which statements to write
- How to measure

3.2 Question Pool

In the Skype meeting, it was agreed that the questionnaire would consist of two sections:

Section 1 would include the questions related to undergraduates' personal information and the level of previous involvement in coding activities.

Section 2 would include 5-point Likert scale, asking the target group to indicate the level of agreement as "Strongly Disagree", "Disagree", "Neither Agree Nor Disagree", "Agree" or "Strongly Agree". APEC prepared an initial Google document including the sections and example statements.

3.3 Further Comments and Compilation of Questions

The academicians of the partner universities contributed to the question pool together with their comments on the sections.

APEC reviewed the comments made necessary corrections or changes. Then the common working document was shared to receive the academicians' opinions once more. A second process was conducted and new comments led to new arrangements. Finally, the academicians of DEU worked on the statements and the sections, and finalized the questionnaire. APEC prepared a mark-up for the questions. All partners used the same mark-up in order to facilitate the data analysis process and provide a standardized format.

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

3.4 Sections of The Questionnaire

Section 1

- Section 1 includes, the titles: Age, Gender, Year of Study, two questions to determine the level of involvement in coding activities and two questions to rate the involvement levels related to some very famous coding programs.

Section 2

- Section 2 includes the statements related to the training needs of undergraduates under the following titles
- Objectives of Coding Education
- Skills and Acquisition in Coding Education
- Content of Coding Education
- Teaching Methods & Techniques That Can Be Used in Coding Education
- Teaching Materials / Tools of Coding Education
- Assessment of Coding Education

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4. Participants / Sampling

A Simple Random Sampling Method was used to select the participants. Partner Universities asked their undergraduates (CS teaching) to complete the questionnaire and undergraduates voluntarily participated in the survey.

4.1 Number of Participants

277 undergraduates from four partner universities participated in the questionnaire using online survey platforms. Surveying period lasted approximately two weeks. However, some of the responses in the data was missing, so only 258 forms were analysed in total. Refer to figure 1 below.

Dokuz Eylül University

Participants: 67

Participants with some missing data: 6

61 participants' responses were used. Participants comprised 23,64% of total.

Limerick Institute of Technology

Participants: 46

Participants with some missing data: 10

36 participants' responses were used. Participants comprised 13,95% of total.

Maribor University

Participants: 38

Participants with some missing data: 3

35 participants' responses were used. Participants comprised 13,57% of total.

Zagreb University

Participants: 126

All responses were used. Participants comprised 48,84% of total.

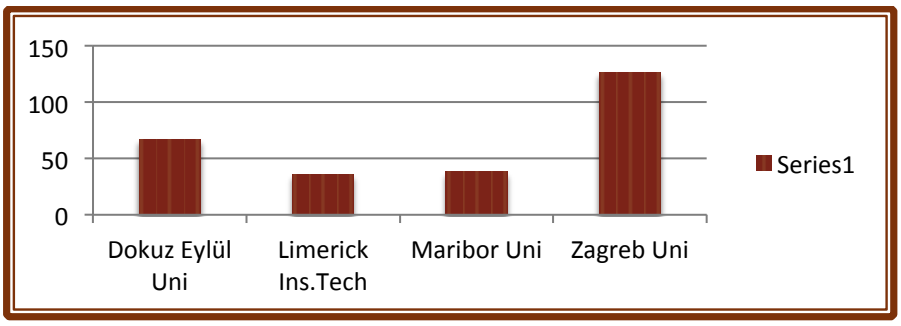


Figure 1 – Number of Participants

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4.2 Age Range of Participants

The age of the participants ranged between 18 - 52. Limerick Institute of Technology has the broadest age range. The age range of participants for each partner university is as follows:

Dokuz Eylül University

Age Range: 20-31

Limerick Institute of Technology

Age Range: 18-52

Maribor University

Age Range: 19-32

Zagreb University

Age Range: 21-32

4.3 Gender

The number of female participants of Dokuz Eylül University, Limerick Institute of Technology and Maribor University represented 30% - 37% of the gender pool, whilst the female participants for Zagreb University presented a slightly higher percentage than their male counterparts as shown in figure 2. In total, the average proportion of participants include 43.4% of female and 56.6% of male, as shown in figure 3.

Dokuz Eylül University

Female: 23 (37,70%)
Male: 38 (62,30%)

Limerick Institute of Technology

Female: 11 (30,56%)
Male: 25 (69,44%)

Maribor University

Female: 12 (34,28%)
Male: 23 (65,72%)

Zagreb University

Female: 66 (52,38%)
Male: 60 (47,62%)

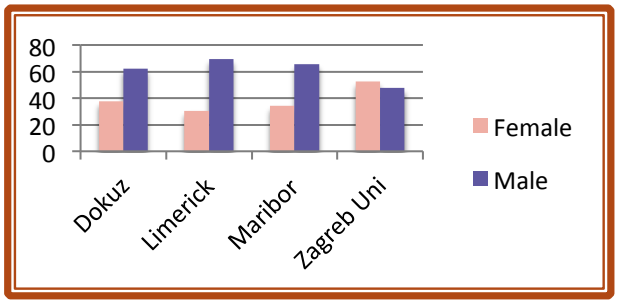


Figure 2 – Gender Distribution by Percentage

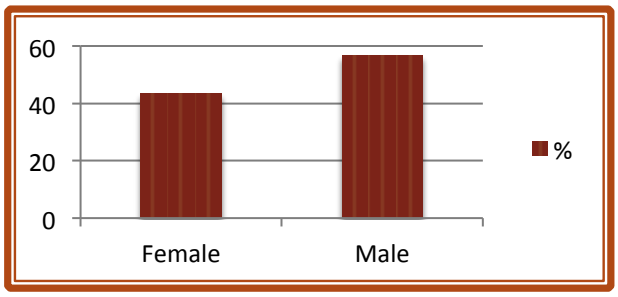


Figure 3 – Gender Distribution by Percentage (of total)

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4.4 Year of Study

The participants from Dokuz Eylül University concentrated on the 2nd, 3rd and 4th years of study. The highest number of participants from Maribor University are completing their 4th year of study, while participants of Zagreb University are mainly in their 4th or 5th year, with the majority of these as year 5 undergraduates. Respondents were mostly experienced undergraduates who could provide data that are more tangible.

Unlike the other three partner universities, 75% of participants of Limerick Institute of Technology are predominantly first-year students. Summaries provided below.

	Dokuz Eylül University		Limerick Institute of Technology		Maribor University		Zagreb University	
	No	%	No	%	No	%	No	%
YEAR 1	0	0	27	75	5	14,28	0	0
YEAR 2	17	27,87	1	2,78	1	2,86	0	0
YEAR 3	18	29,51	1	2,78	7	20	3	2,38
YEAR 4	18	29,51	6	16,66	14	40	43	34,13
YEAR 5	8	13,11	1	2,78	5	14,28	80	63,49
YEAR 6	0	0	0	0	3	8,57	0	0

Table 1 – Year of Study by Numbers and Percentage (of Each Partner University)

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Number	32	19	29	81	94	3
%	12,4	7,36	11,24	31,4	36,43	1,17

Table 2 – Year of Study by Numbers and Percentage (of Total)

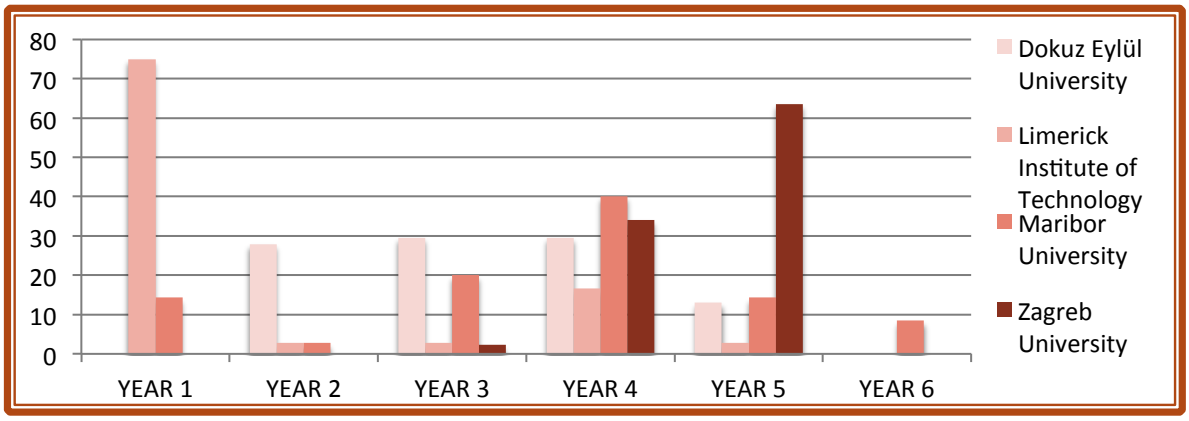


Figure 4 – Year of Study by Percentage (of Each Partner University)

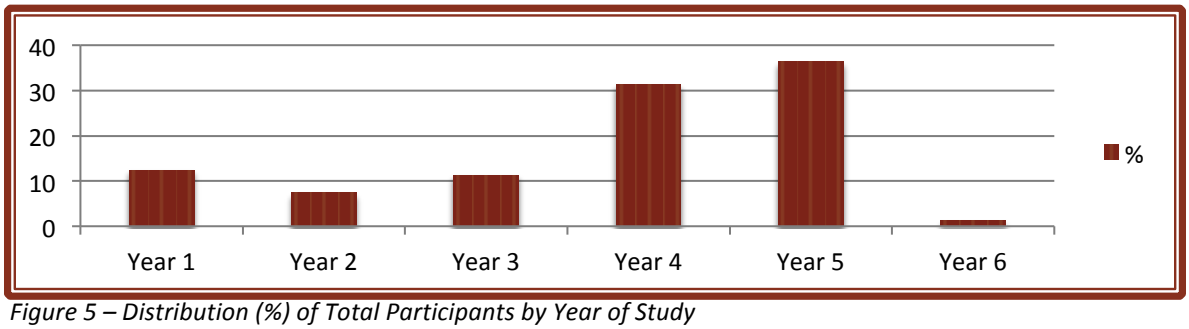


Figure 5 – Distribution (%) of Total Participants by Year of Study

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4.5 Coding Awareness

The coding awareness amongst participants of Zagreb and Maribor Universities is very high, especially in Maribor University where nearly all of participant undergraduates already knew about coding. 67% of participants from the Dokuz Eylül University also indicate awareness of coding and 78% of the participants from Limerick Institute of Technology indicated that they were also aware of coding activities in education.

	Dokuz Eylül University		Limerick Institute of Technology		Maribor University		Zagreb University	
	No	%	No	%	No	%	No	%
Already Knows	41	67,21	28	77,78	34	97,14	107	84,92
Doesn't Know	20	32,79	8	22,22	1	2,86	19	15,08

Table 3 - Coding Awareness by Number and Percentage (of Each Partner University)

	Already Knows	Doesn't Know
Number	**210**	**48**
%	**81,4**	**18,6**

Table 4 - Coding Awareness by Number and Percentage (of Total)

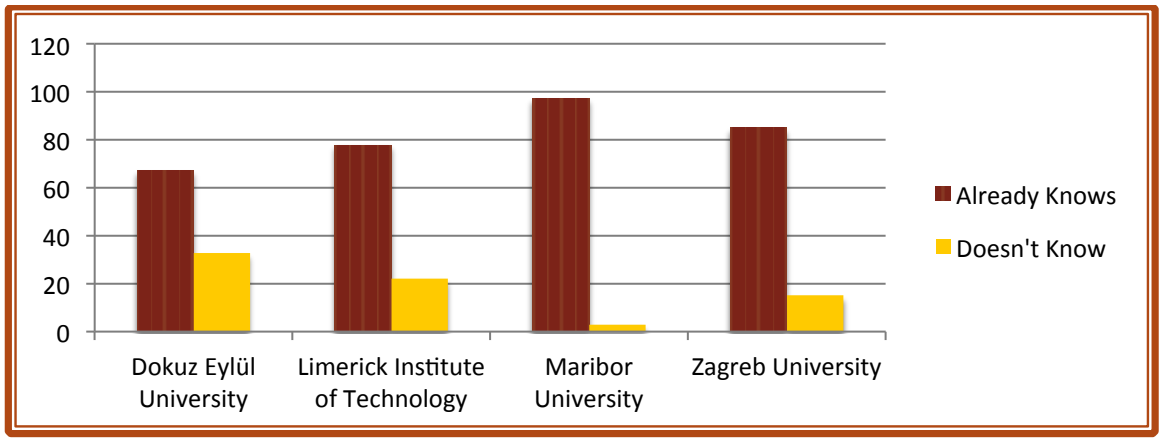


Figure 6 – Coding Awareness by Percentage (of Each Partner University)

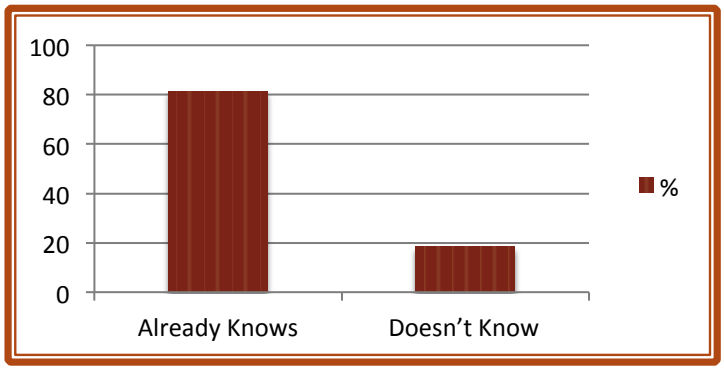


Figure 7 – Coding Awareness by Percentage (of Total)

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4.6 Involvement in Coding Activities

The rates of involvement in coding activities were also very high amongst the participants. As expected, the results were very similar to the responses to the previous question. There are several online coding platforms teaching children on how to code. Becoming involved in these practices most likely will increase the awareness of coding in education.

	Dokuz Eylül University		Limerick Institute of Technology		Maribor University		Zagreb University	
	No	%	No	%	No	%	No	%
Involved	38	62,3	20	55,56	32	91,43	118	93,65
Wasn't Involved	23	37,7	16	44,44	3	8,57	8	6,35

Table 5 – Involvement in Coding Activities by Number and Percentage (of Each Partner Uni.)

	Involved	Wasn't Involved
Number	**208**	**50**
%	**80,62**	**19,38**

Table 6 – Involvement in Coding Activities by Number and Percentage (of Total)

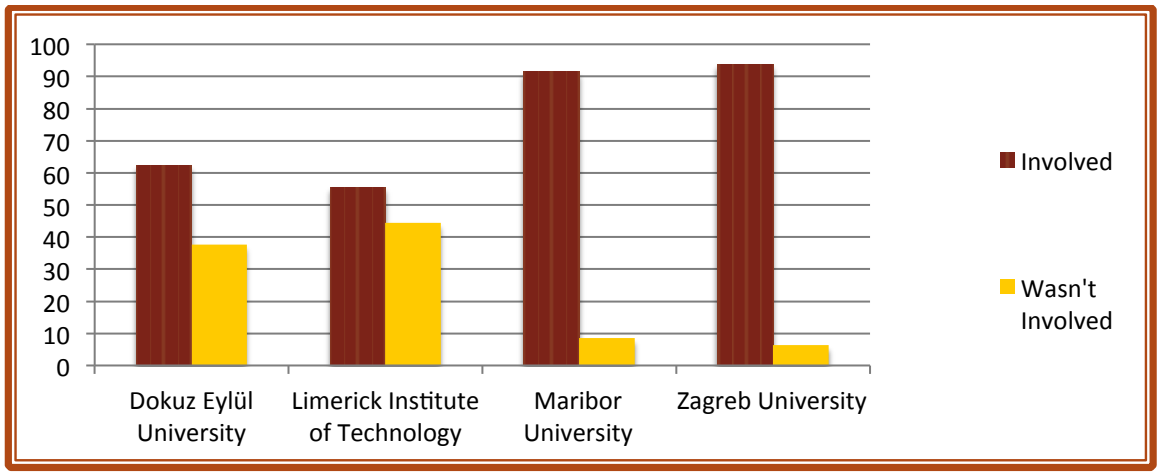


Figure 8 – Involvement in Coding Activities by Percentage (of Each Partner University)

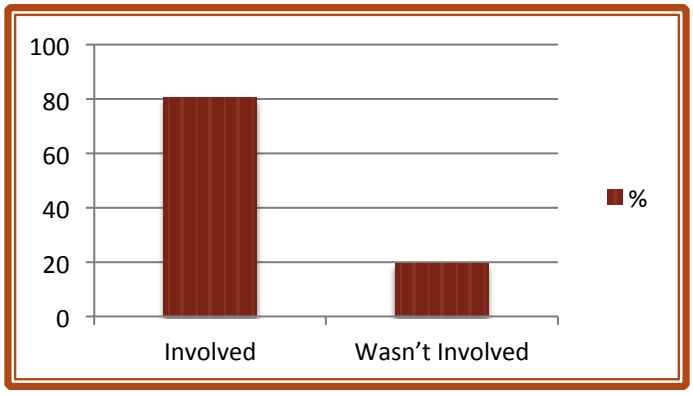


Figure 9 – Involvement in Coding Activities by Percentage (of Each Partner University)

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4.7 Which Coding Programs are more familiar?

As a result of the general survey analysis, Educode observed that:

"Blockly" and "Alice" are the most unknown coding programmes among the participants. 84 participants of the Zagreb University indicated that they have never heard of Blockly and likewise 82 for Alice. 23 Participants form Limerick Institute of Technology on the contrary were involved in Blockly but 30 of them had never heard of Alice.

A relatively high number of participants from Dokuz Eylül University also indicated that they had only heard of Phyton (36) and Robotics (28). Similarly a high number of participants of Maribor University indicated that they had never or only heard of C/C++ (26), Java (17) and Robot Programming (29). The participants from Limerick Institute of Technology indicated that Python (22) and C/C++ (19) were mostly unknown, whilst Java (25) was very familiar.

91 participants from Zagreb University indicated they were regularly involved with Python. 50 participants of Dokuz Eylül University indicated that they were very involved with C/C++, and 33 participants of Maribor University indicated they were very involved with Scratch. Distribution varies quite substantially between the participants of partner universities.

4.8 Which coding programmes are more attractive?

As a result of a general analysis, Educode observed that:

Participants of Dokuz Eylül University more inclined to learn Python, Java and Robot Programming, Blockly and Alice were the most unattractive coding programmes amongst these participants, while Scratch attracted a little more attention.

Participants of Limerick Institute of Technology mainly use three coding programmes, which are Scratch, Blockly and Robot Programming. Interest was shown for use of Python, but Java and C/C++ didn't attract interest as they indicated they already knew how to use these programmes. Alice wasn't attractive at all.

Participants of Maribor University inclined more to Java and Robot Programming. Python and C/C++ also caught their attention. The participants showed no interest in Alice and Scratch but showed slight interest in Blockly.

Participants of Zagreb University were more inclined towards Blockly, Scratch, Alice and Python. Even "all" participants indicated that they wanted to learn Alice. Only half the participants showed interest in C/C++ and Java. 86 participants indicated that they didn't want to learn Robot Programming.

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

5. Data Analysis

Section 2 is to present the findings of the training needs of the participants from the questionnaire. Educode observed that participants needed trainings for coding education at a very high level. Since the rates of training needs were very high among participants, Educode mainly compared small differences between the responses.

- Frequencies for each item were analysed and data sets were formed.
- Each country's data set included the number of responses so as to display the level of agreements from 1(strongly disagree) to 5 (strongly agree).
- Data, displaying level of agreements (from 1 to 5) were placed one under the other. The uppermost numbers represented the strong disagreement while the under-most numbers represented strong agreement.
- For example, for each statement, we could see how many participants strongly disagreed, disagreed, neither agreed or disagreed, agreed, or strongly agreed in each partner country's data set.
- The total frequencies were considered to reveal general training needs of all participants, but country specific needs were also determined in order to address all training needs and also to ensure the transferability of the training programme.
- Mode Values were used to clarify the most frequently preferred choices for each part in section 2.

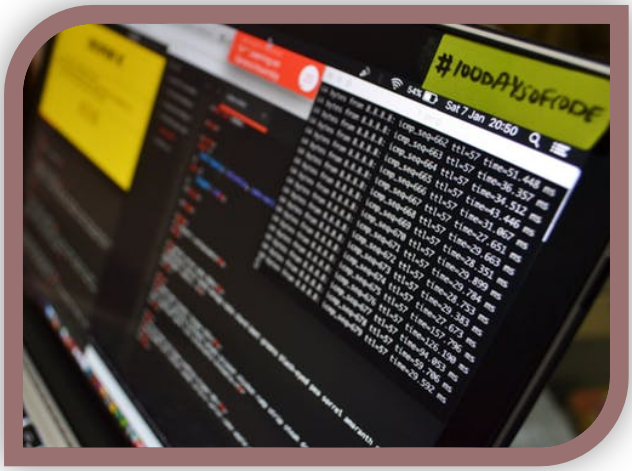
Curriculum Components of Coding Education

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

6. Findings

6.1 Objectives of Coding Education

Most of the participants strongly agreed or agreed (Mode: Strongly Agree) that they needed training on the objectives of coding education. Whilst the main drivers of coding education, EU and national policies about coding education and contributions to coding education for interdisciplinary studies were slightly less attractive topics to the participants. Interestingly, one third indicated that they didn't need or weren't sure that they needed to learn EU and National policies about coding education.



All participants of Dokuz Eylül University indicated that they needed to learn objectives of coding education. Only one sixth of participants weren't sure that ICT & STEM Teachers were particularly critical to the successful delivery of coding education (Statement 4).

Almost one third of the participants from Limerick Institute of Technology were unsure whether they needed to learn the main drivers of coding education, coding education for interdisciplinary studies and also the critical role of ICT/STEM teachers for the successful delivery of coding education (Statements 1, 3 and 4). Whilst all statements attracted a great attention in this section, basic concepts of coding education, basic strategies of problem solving in the context of coding education and evaluating different solutions of the same problems and distinguish between good and bad ones (Statements 5, 7, and 8) were the most attractive subjects in this section.

Participants from Maribor University showed great interest to all statements in this section, with the exception of first two questions where one fifth of participants indicated they didn't need to learn the main drivers of coding education and relevant EU and National policies and initiatives supporting coding education (Statements 1 and 2).

Participants from Zagreb University wished to concentrate on the needs regarding successful delivery of coding, problem solving and computational thinking (Statements 4, 6 and 7). From this point of view their answers were similar to the participants of Maribor University.

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

6.2 Skills and Acquisition of Coding Education

Most participants strongly agreed or agreed (Mode: Strongly Agree) that they needed training on the skills and acquisitions of coding education. Students were interested in which skills they could develop through completing the coding programme.

All participants of Dokuz Eylül University indicated that they needed this education. They were most attracted to topics for the teaching of simple algorithms and using different data structures to store data in appropriate ways (Statements 5 and 6).

Most of the participants (Two thirds) from Limerick Institute of Technology indicated that they needed to learn all the content in this section. However one fourth of participants were unsure that they needed to learn which type of skills students can develop through completing coding courses, how to translate verbal description of domain problem into formal form, how to abstract (or simplify) the problems' solution while students acquire related skills. (Statements 1, 3, 4, and 6). 88.9% of the participants wanted to learn how to transfer their IT skills to students and modeling techniques for successful programming (Statements 2 and 7).

Participants of Maribor University were also very willing to take training on this topic. Only one fifth of participants weren't sure that they needed to learn how to teach to translate verbal description of domain problem into formal form and modelling techniques for successful programming (Statements 3 and 7).

Participants of Zagreb University also indicated that they needed to take training in this topic. The average percentage of participants who weren't sure that they needed to learn this topic was 11.1%. The statements showing greatest interest were; which type of skills students could develop through completing coding courses and using different data structures to store data in appropriate ways (Statements 1 and 4).

6.3 Content of Coding Education

Generally, most of the participants were also interested in taking training on this topic (Mode: Strongly Agree). All the statements in the scale were rated as strongly agree, however 9% of the participants weren't sure that they needed to learn how to teach the differences between procedural and object orientated programming (Statement 8).

Participants of Dokuz Eylül University mostly indicated that they needed training on the topic. Yet one tenth of participants weren't sure that they needed to learn which topics relevant to procedural programming (Statement 3).

Three quarters of the participants from Limerick Institute of Technology indicated that they needed to take training about the content of the coding education. The most preferable subject was to understand authentic problems relevant to introductory programming (Statement 4).

Participants of Maribor University needed training on this topic. But one quarter of participants were not sure or didn't want to learn how to teach the difference between Static (C/C++, Java) and Dynamic (Python) languages and the differences between procedural and object orientated programming (Statements 7 and 8).

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

85.71% of the participants of Zagreb University wanted to take training on this topic. Their interests were focused mostly on authentic problems relevant to introductory programming, debugging and difference between Static (C/C++, Java) and Dynamic (Python) languages (Statements 4,5 and 7).

6.4 Teaching Methods & Techniques of Coding Education

All the participants indicated that there is a training need in general on this topic (Mode: Strongly Agree). The most prominent training needs indicated are; new methods of how to engage students in the topic of coding, how to engage students with different abilities and learning styles, how to apply project-based learning in coding, how to apply game based learning in coding, how to identify which problems are more suitable to motivate students for programming, and suitable online platforms providing video examples in coding (Statements 2, 3, 4, 7, 12, and 19).

Slightly less attractive training needs include how to apply flipped classroom in coding, how to apply blended learning in coding, how to use problem presentation (preparing requirement specification) as a good warm-up method, how I could use competition between groups as warm-up, how to use graphic techniques (flow chart, class diagram) that help in understanding of source code of some program (Statements 5, 9, 16, 17, and 18). 20% of the participants were not sure or didn't need to learn about teaching the history of computer science (programing languages development) as part of coding education.

Participants of Dokuz Eylül University indicated that they needed this training. They showed interest on how to identify which problems are more suitable to motivate students for programming, how to use programming both on-line and offline and case studies, example problems/solutions more (Statements 12, 19 and 20). One tenth of participants were not sure that they needed to learn flipped classroom and inquiry-based learning to teach coding (Statements 5 and 8).

73.8% of participants of Limerick Institute of Technology indicated that they agreed that they needed to learn teaching methods and techniques of coding education. Particular interest was indicated in the topics how to apply project-based learning in coding, how to apply problem-based learning in coding, how to apply game based learning in coding, how to identify which problems are more suitable to motivate students for programming, and how to use programming both on-line and offline (Statements 4, 6, 7, 12, and 19). The history of computer science (programing languages development) as part of coding education and examples of easy tasks/projects, which they can give their students were the least attractive subjects (Statements 14 and 15).

Interest from the participants of Maribor University were focused on, the needs related to lesson plans, new methods, learning styles, game-based learning, motivational tools, algorithms, how to identify suitable problems, examples of easy tasks and projects and programming both off-line and online (Statements 1, 2, 3, 7, 10, 11, 12, 15 and 19). However

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

one third of participants were not sure or didn't want to learn on how to use problem presentation (preparing requirement specification) as a good warm-up method (Statement 14).

Participants of Zagreb University also required this training. Yet blended learning, history of computer science, problem presentation as a warm-up method, competition between groups as warm-up, and the usage of graphic techniques were slightly less attractive topics compared to others (Statements 9, 14, 16, 17, and18). More interest were shown in the topics, new methods and learning how to identify which problems are more suitable to motivate students for programming (Statements 2 and 12).

6.5 Teaching Materials and Tools of Coding Education

Teaching materials and tools of coding education are one of the most attractive topics in the questionnaire (Mode: Strongly Agree). Especially "how to use tools and ways to increase students' confidence about coding topics, source code and working examples of projects, students can use, and suitable online platforms providing video examples in coding (Statements 1, 3, and 8) were the most intriguing issues among participants.

Participants from Dokuz Eylül University wish to learn this topic. The most focused needs were on how to use tools, useful websites, and collaboration with other teachers, and online platforms providing video examples (Statements 1, 2, 4, and 8).

80% of participants from Limerick Institute of Technology indicated that they needed to learn about teaching materials of coding education. The most attractive training topics were useful websites to teach coding more effectively, source code and working examples of projects my students can use, good IDE, and suitable online platforms providing video examples in coding (Statements 2, 3, 6, and 8).

Participants from Maribor University also needed to receive training on this topic. All topics attracted the participants but one fifth of them didn't need to learn where to collaborate with other teachers (Statement 4) and one third of them didn't need to learn about Integrated Development Environments (Statement 6).

Most of the participants from Zagreb University also wanted to receive training on this topic. One tenth of participants on average weren't sure if they needed to learn this topic. The topics with most interest were how to use tools, working examples, high quality free e-textbooks and resources, online platforms providing video examples (Statements 1, 3, 7, and 8). The last one (8) being the most attractive topic amongst them.

6.6 Assessment of Coding Education

Most participants needed training about assessment of coding education (Mode: Strongly Agree). They needed to learn how to know if their teaching was to be successful and targeted skills were to be acquired. However, how to use summative assessment techniques, how to use criterion and norm based assessment, and different types of questions in CS education (Statements 4,6, and 10) were slightly less attractive as subjects. Almost 25% of the participants

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

indicated that they were not sure or didn't need to learn how to apply Revised Bloom's taxonomy to the coding.

Participants of Dokuz Eylül University indicated that they needed training on this topic. Yet one tenth of participants were not sure that they needed to learn about summative assessment, criterion and norm-based assessment and Bloom's Taxonomy (Statements 4, 6 and 9). Different assessment methods and how to check plagiarism were most welcomed topics (Statements 1 and 2).

It is identified amongst the participants from Limerick Institute of Technology that the topic, training need as the assessment of coding education was of least interest. One third of the participants either didn't want to learn or wasn't sure that they needed to learn about assessment of coding education. While the most attractive topics were peer assessment in the context of coding and how to evaluate code (Statements 5 and 8), the least attractive topics were how to use summative assessment techniques, how to use criterion and norm based assessment, and how to apply Revised Bloom's taxonomy to the coding (Statements 4, 6 and 9).

Participants of Maribor University needed training on this topic, yet one fifth of participants were not sure that they needed to learn peer assessment and Bloom's Taxonomy (Statement 5 and 9).

Nearly 80% of participants from Zagreb University indicated that they needed to learn this topic. They mostly focused on how to check plagiarism and different question types (Statements 2 and 8). Nearly 20% of the participants were not interested in the types of assessments, Bloom's taxonomy and different types of questions (Statements 3, 4, 5, 6, 9 and 10).

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

7. Conclusion

As a result of the training need analysis, TNA, the partnership of Educode are recommending the preparation of a training programme (Curriculum) concentrating on "objectives of coding education", "skills and acquisitions of coding education", "teaching methods and techniques", and "teaching materials and tools". Other sections should be considered, from a superficial perspective, since there are statements that one needs to teach another.

Coding Education has become a necessity for new generations in order to have 21st century skills, pursue interdisciplinary studies and participate in the qualified workforce. In 2014, England introduced programming to children from the age of five. Austria included coding education in its "New Digital Technologies Curriculum" starting form the age of seven in 2015. Many countries such as India, Malaysia and USA have started to give a place to coding education in their curricula. There have been serious initiatives to prepare teachers for coding education recently. For example, code.org (US based) prepared 495,000 new teachers to teach computer science for the grades K-12 in 2016.

Candidate ICT/STEM teachers need to know the main drivers of coding education, why it is so critical and what kind of a perspective they should have in order to see the global picture and upcoming opportunities. They also need to know what skills; acquisitions and content are appropriate for children between the ages 11-17 to have a good comprehension of the subject. Learning teaching methods and techniques, and different practices, which can facilitate children to code more easily, is the most critical point, which will turn efforts to success during the course delivery. Well-determined teaching materials and tools will provide new teachers to help their students become engaged in the programming activities and start producing suitable qualified people. Finally, they need to know effective ways of assessment in coding education. In this way, they will be able to evaluate to what extent learning takes place and its quality.

Since the duration of the training will be limited, the content of the training should be more focused and for urgent needs. When the total responses were analysed, observed that participants agreed to these statements more than the others:

1- I would like to know about computational thinking. **(OBJECTIVES)**
2- I would like to know basic strategies of problem solving, in the context of coding education. **(OBJECTIVES)**
3- I would like to know how to evaluate different solutions of the same problems and distinguish between good and bad ones. **(OBJECTIVES)**

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4- I would like to learn which skills students can develop through completing coding courses (SKILLS)
5- I would like to transfer my IT (Information Technologies) skills to my students. (SKILLS)
6- I would like to understand authentic problems relevant to introductory programming. (CONTENT)
7- I would like to learn how to apply game based learning in coding. (METHODS)
8- I want access to source code and working examples of projects my students can use. (MATERIALS)
9- I would like to know suitable online platforms providing video examples in coding. (MATERIALS)

It was observed that participating undergraduates focussed more on the delivery process of coding education and that they were also more interested in practical solutions for successful delivery.

The most preferred needs under other sub-sections recommended to be included in the training program Educode as follows:

1. Objectives of Coding Education

Recommended training topics under this title are as follows.

1- Computational thinking
2- Basic concepts of coding education
3- ICT & STEM Teachers' critical role for the successful delivery of coding education
4- Basic strategies of problem solving, in the context of coding education
5- Handling different solutions of the same problems and distinguish between good and bad ones

In addition, main drivers of coding education and its importance in disciplinary studies can be handled as well.

2. Skills and Acquisitions of Coding Education

Recommended training topics under this title are as follows.

1- Which skills students can develop through completing coding courses
2- How to transfer IT (Information Technologies) skills to students
3- How to use different data structures to store data in appropriate ways
4- Implementing simple algorithms to teach coding
5- How to abstract (or simplify) the problems' solution while students acquire related skills
6- Modeling techniques for successful programming (coding)

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

In addition, how to translate verbal description of domain problem into formal form (pseudo language, and etc.) can be handled briefly during the courses.

3. Content of Coding Education

Recommended training topics under this title are as follows.

1- Detailed content, which supports delivery of coding lessons
2- How to teach basic algorithms, sequences, loops, and other methods used within coding
3- Topics relevant to procedural programming
4- Authentic problems relevant to introductory programming
5- How to teach debugging as an important part of coding
6- How to teach testing as an important part of coding
7- The difference between Static (C/C++, Java) and Dynamic (Python) languages

In addition, "How to teach the differences between procedural and object orientated programming" can be handled briefly during the courses.

4. Teaching methods and Techniques of Coding Education

Recommended training topics under this title are as follows.

1- How to prepare lessons plans for relevant coding topics
2- Methods of how to engage students in the topic of coding
3- How to engage students with different abilities and learning styles
4- How to apply project-based learning in coding
5- How to apply problem-based learning in coding
6- How to apply game based learning in coding
7- How to apply inquiry-based learning in coding
8- Motivational tools (CS unplugged, pedagogical games, algorithm visualization) to engage students in coding education
9- Step by step execution of algorithms as motivational tool
10- Problems more suitable to motivate students for programming
11- Specific video/movie/cartoon segments which can be used to in coding education
12- Lots of examples of easy tasks/projects
13- How to use programming both on-line and offline
14- Specific content e.g. Case studies, example problems/solutions etc. to use directly in teaching

In addition, how to apply flipped classroom in coding, how to apply blended learning in coding, how to use problem presentation (preparing requirement specification) as a good warm-up method, competition between groups as warm-up, how to use graphic techniques (flow chart,

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

class diagram) that help in understanding of source code of some program can be handled briefly, if allocated time for the course is adequate.

5. Teaching Materials and Tools of Coding Education

Recommended training topics under this title are as follows.

1- How to use tools and ways to increase students' confidence about coding topics
2- Useful websites to teach coding more effectively
3- Source code and working examples of projects my students can use
4- Finding out about where we can collaborate with other people teaching coding
5- How to adapt learning scenarios for my lessons plans
6- Good IDE (integrated development environments).
7- How to find high quality free e-textbooks and resources for coding
8- Suitable online platforms providing video examples in coding
9- How to use modern mobile applications (smartphone, tablets (e.g. Kahoot, Socrative and book widgets for instance) to review a lesson at the end in class

6. Assessment of Coding Education

Recommended training topics under this title are as follows.

1- Different assessment methods for coding education
2- How to check for plagiarism in the context of coding education
3- Formative assessments as a continuous form of providing feedback to students
4- How to use peer assessment in the context of coding
5- Different software tools which can support assessment
6- How to evaluate code (line of codes; logical paths; … different code metrics)

In addition, how to use summative assessment techniques, how to use criterion and norm based assessment, different types of questions in CS education can be handled briefly during the courses.

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

The recommended sections for the curriculum are as follows.
MAJOR PHILOSOPHY OF THE PROGRAMME
GENERAL OBJECTIVES OF THE PROGRAMME
KEY COMPETENCES & CONDITIONS
- FOR LECTURER
- FOR TRAINEES
SUGGESTED TESTING TECHNIQUES FOR THE ASSESSMENT OF AQCUIRED SKILLS
STRUCTURE OF THE CURRICULUM
CODING EDUCATION CURRICULUM
- OBJECTIVES
- CONTENT
- TEACHING METHODS & TECHNIQUES
- TEACHING MATERIALS & TOOLS
- ASSESSMENT
TEACHING PRACTICE RECOMMENDATIONS
LESSON PLANS

Finally, in order to determine the effectiveness of the trainings, it is recommended that pre and post-tests are carried out.

Also short interviews, meetings with instructors and training participants and observation notes would be useful for formative assessment, which will enable Educode to conduct in-depth analysis of the training processes.

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Appendix 1 – Interview Form
Appendix 2 – Mark-up
Appendix 3 – Data Sets
Appendix 4 – Infographic of the need analysis
Appendix 5 – Template Curriculum

This project has been funded with support from the European Commission. This presentation reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.