**EU Report: Computational thinking in education**

**Speaker 1** [00:00:00] Welcome to EU Code Week Podcasts. We bring coding, computational thinking, robotics and innovation closer to you, your community and your school.

**Speaker 2** [00:00:20] Hello everyone and welcome. We are your hosts Eugenia Casariego and Arjana Blazic.

**Speaker 3** [00:00:24] the EU code with team is here to help teachers bring coding and programming into the classroom by providing the latest training, resources and materials, as well as insights on current and future implementations and best practises in order to transform education.

**Speaker 2** [00:00:43] Indeed. And this time we will talk about computational thinking and computational skills and their relevance to education field in Europe.

**Speaker 3** [00:00:50] To shed some light on the current progress of integrating computational thinking into the curriculum and what needs to be done to improve. Today's episode will be about the recent research study announced by the European Commission reviewing computational thinking in compulsory education, state of play and practises from the field. Talking us through the report and its findings are today's guests Katja Engelheart and Augusto Chiocarello. Katja is a senior education analyst that European Schoolnet with a keen interest in making school education policies more inclusive and innovative. While Augusto is a researcher at the National Research Council of Italy interested in educational technology and STEM education.

**Speaker 2** [00:01:37] Hello to you both and welcome to the EU Code Week podcast.

**Speaker 3** [00:01:40] Yeah, thanks a lot for inviting us. We are both very happy and excited to be here. It's our first podcast, so we're very happy to share the findings of our study with you today.

**Speaker 5** [00:01:49] Thank you very much for joining us today. Now focussing on the topic at hand, can you tell our listeners briefly what the scope of the study was.

**Speaker 4** [00:01:58] The commission is interested in what is computational thinking? How the different countries start addressing computational thinking, in their curricula? And so we were asked to investigate this subject and then to find out actually what's going on because this is quite new and we wanted to know from the field.

**Speaker 5** [00:02:16] Knowing that this current 2022 Computational Thinking in Compulsory Education Study is actually a follow up of an earlier 2016 report, which was also focussed on the same topic. What are the main similarities and differences between the two studies?

**Speaker 5** [00:02:34] Back in 2016, when we did the first report, computational thinking was still a very new and a bit the hottest new thing hype to look at. So the conclusion of that first report was in a way that computational thinking is a promising concept. And now with our second edition of the study, which was published in February 2022, the conclusion really was that it's more than a promising trend now that it has arrived, at least in curricula across Europe. And of the 29 European countries that we analysed, 25 already introduce basic computational thinking or computer science concepts in their curricula and one way or the other and the other four countries, either pilot activities or plan to do some things, so we can really say, you know, what has arrived in the curricula and as a very general trend, you can also say that it goes towards integrating computer science concepts also at more grades and more levels, so more comprehensively and more countries are switching from offering computer science concepts only as only as optional subjects or optional activities to really also make it compulsory so more likely to reach more students.

**Speaker 2** [00:03:44] Can you clarify to the listeners which of the countries where involved in the latest one or on the whole EU?

**Speaker 5** [00:03:50] Was focussed on 29

**Speaker 4** [00:03:52] There are some countries that we specifically addressed and there were nine. We had a good overlap that the interesting things was we had many more countries replying to this survey than to the previous one.

**Speaker 2** [00:04:03] That's good!

**Speaker 3** [00:04:04] In only five years there have been so many substantial changes in teaching computational thinking and computer science in schools all over Europe. What is the current situation and what are some of the challenges for computer science education in Europe?

**Speaker 4** [00:04:20] We really had the chance to see many countries having the policy in place, so they decided to do it and you have the curriculum as a framework reference. What actually is seen on the big issues, how you implemented the decision. And in that way, the differences in the national context is huge. The big issues in those countries, this thing is quite new and you cannot hire teachers, so you have to deal with the teachers you have, the this one is one major issues, but then you have specific things. You know some countries have decided to do it having their own particular subject, others they do within other subjects. So many do within maths for example, and so you have a maths teacher now has to do another complete new section.

**Speaker 2** [00:05:11] Coming back a bit more towards the meta-level (?) Of today's podcast. So very briefly, can you tell us more about the study methodology? So mostly, which were the main inquieries you set to to answer and the key terms used in the report.

**Speaker 4** [00:05:23] So when we designed the study, we try to use a number of different means. So we try to triangulate this idea of how computational thinking is addressed into school with different means. We did a literature review, just to make sure that we had a good understanding of what was the definition around and what the literature was telling us about the number of issues. One big major effort was a survey, so we came up with an update of our previous questionnaire that was sent to the Ministry of Education. Just to understand, you know, broadly how this subject was going on. Then we decided to do an in-depth case study, but we could not afford to do with many, and we did with nine. And the idea was to address different issues at the school level. Primary level, lower secondary, but also the differences in the way is addressed you know, as a separate subject or within other subjects, in many cases is math. And if you want to be very specific on the primary level, we did in Norway, Lithuania and Slovakia. Then we did another two sets of three countries, one where things are done, you know, within a subject specific. And there was England, then Poland and Croatia. And then the final one was about, you know, when it's dealt with in mathematics and technology. And then we looked at France, Finland and Sweden. We also held two excellent workshop in the first one was, you know, people experience in the field from Europe and outside. And the second was sort of a validation workshop. We had, you know, a preview of our findings and we see what the reactions were before we wrote the final report.

**Speaker 2** [00:07:09] I find this especially interesting, and I guess so for our listeners in the comparison between the case studies, for each of these case studies, what were the main similarities and differences that you found in how computational thinking is addressed for each case and what are the key findings?

**Speaker 5** [00:07:23] So I looked specifically into Croatia, England and Poland myself, and we carried interviews out there with policymakers, experts, school heads, teachers and students and their specifically. So that's lower secondary education. We started with the assumption that there might be a difference between Poland and Croatia already having a long tradition of computer science compared to England, where that's really not the case. We did not really find such conclusive evidence for quite practical reasons in a way, because the subject is already there integrating a new subject in any curriculum is very different, initial teacher training is already been in place and some teachers trained in the subject already there. So that's already quite a lot. But still also the curriculum changes in both countries. Poland, Croatia were quite extensive. So really integrating a lot of new grades, making it compulsory, making quite a lot of change also on a curriculum. So nonetheless it was also a big challenge in both countries and teacher training was required and we looked specifically in this case study at how computational thinking is integrated as a separate subject. And I guess the most interesting thing we found there, that it, normally this computer science subject does not only cover computer science concepts and computational thinking activities, but in all cases also more generally user digital skills, how to be safe online and all these things. And so it becomes relevant how to link those two areas of general skills, because we all agree all those digital skills you need, but it also becomes a challenge, then, in practical terms, how much time is actually in reality dedicated to computational thinking activities?

**Speaker 4** [00:09:01] In a way we were interested in to see how this integration within within another subject would really work. And we actually find out there was more than one subjects. Initially, we we had the initial assumption because we did try to compare the case study also triangulate it within the different countries. We had the impression on paper beginning that maybe Sweden was going to be a little bit different because of their tradition of digital competences and because they in the curricula, they also include civic studies. So we said, wonderful, there is not just STEM, there is just some humanistic discipline involved. We also had, you know, some sort of inside question. If you do it within mathematics, do you want to improve the learning of mathematics crossing with computational tools. Well, the nice thing about, you know, doing experimental studies, is that you don't find what you're looking for. But at the end, if you look at the curriculum it's similar, mathematics in those three countries takes the big burden of introducing basic computer science concepts. In addition to the mathematics curriculum. In the technology, you do more the practical part. So you do the robotic part, you construct some things. And the main differences that we found, was how teacher training is handled because in some cases it's compulsory. So teachers have to do their professional development but then you need to find substitute teachers or like in France is not compulsory. So you have to find other ways to address these things. Because mathematics is such an [00:10:32]icetake? [0.0s] subject, it's always in the final exam. So in France, one of the question on the final exam for mathematics has to do with computer science. And in some programming, Finland is an exception. They don't have the final exam, but they have very specific instructions for how to assess what the students have learned in programming. But in all those countries, the interesting finding is that they had a very concise curriculum and the curriculum there is very [00:11:03]said/Set?, [0.0s] because they have teacher autonomy and all these nice things. But over the year, in order to support this implementation, the government starts to produce more and more guidelines, prescriptions, supportive materials, so that things become a little bit more explicit. And this is a trend.

**Speaker 2** [00:11:21] Did you also find a lot of differences in how national governments and ministries more in particular provided that teacher training? Would you say that most countries are now up to date with providing teacher training in computational thinking or just more generally in ICT skills?

**Speaker 4** [00:11:35] They all do it. The problem is, what are the policies of the national level. The differences are, and I think for example, in England they started saying, you know, this is an easy things. We are going to invest £2 million and everything is going to work fine and they end up investing £80 million because training a lot of teachers, when you make compulsory it's not just one, a few teachers, it's a lot of them and it becomes a big issue. And for example, when we interviewed the French Ministry of Education, they consciously said we are not going to address primary first, because if you address primary, it will cost us a lot of money. We will start with lower secondary because we assume that the maths teacher are close enough. So yes, we need to put an effort, but maybe it's not such a big effort, so we can start and it's going to be a long process, and it's not going to be easy and it's going to it's going to be expensive I think, you know. But they realise it's a long process.

**Speaker 5** [00:12:31] Yeah. I also wanted to add, I guess in general there are more similarities than differences between the countries. So teacher training is really the main challenge in all countries still. And if you think about it, it is a crazy thing, in a way to integrate something that is entirely new towards the curriculum, that teachers have not been trained for. That is difficult and complicated and to start that curriculum before you actually trained all your teachers to do it. But it's very difficult also to do differently because it's difficult to get funding for teacher training for a subject, or content that doesn't exist. But I think we have to acknowledge that we really put teachers in Europe in front of a challenge there. And what you can see, I mean, England as an example, but also other countries that there is more teacher training available, that there's a lot of different offers, online, offline, training materials that you can use yourself that's really different formats. So that's good. So in a way, the gap has been closed to some extent, which especially in the beginning was also a lot of grassroots movements that stepped in and provided that sort of support to teachers. And now I think the really interesting question is, are those of us that maybe sometimes also pragmatic, really what we need to give teachers all the skills they need. And in the end, how do we know that what we teach the teachers to then teachers in the classroom is actually what we want to reach with our curriculum? You know, we need money. We need a lot of stakeholders also to work together. CodeWeek is one example helping teachers, but there's a need for more support for the teacher. And you can't expect the teachers to do this big challenge on their own.

**Speaker 2** [00:14:05] Thank you, I think I think those are very valid questions, but indeed a very, very good way to put it, that we cannot really expect teachers to do it on our own. But as always, when we provide teacher training, especially in a new field, it's a huge adjustment. And it takes even, can I say even years in plural, just getting to the right point and adjusting and updating it all continuously. So so that's very true of very valid questions. Now stepping,a bit out of teacher training and more into practise, which is what Augusto was also talking a bit right now, how actually countries go about teaching computational thinking and computational skills. You mentioned that there's more similarities in a way than differences and now you provided a few examples of how teachers go about it. So what are some common concepts or strategies used across the curriculum to teach these skills?

**Speaker 5** [00:14:49] For me, the main starting point is always Why do countries integrate computational thinking? And there you see a lot of commonalities. So in general, they, all countries do it for the same reasons because they want to foster these logical thinking skills, problem solving skills. So there is an assumption that students need these kind of skills for their future lives, both professionally and personally. And then what emphasis they will put between those schools depends a bit. Some countries will also have a bit more specific rules, like, for example, attracting students to mathematics studies. So it's generally the same reasons, but then there will be specifics in the countries. And that's I would also say always the starting question why do I do this? Where do I want to go? And from there, countries should start designing their curricula. And maybe just a few words because I hope teachers are listening to us. We do not have conclusive evidence on how teachers do it in the classroom, precisely. But we talk to example schools because often the curricula are quite open, which is in a way also nice. So there's a lot of autonomy for teachers to decide what they want to do. And a lot of times what seems to lend itself to teaching computational thinking very well, it like these, what we call active pedagogical approaches, so really collaborative personalized learning, project based learning, basically everything that helps the students to learn autonomously, to work on their own project, on their own tasks, to think for themselves, to try to find their own mistakes. And in, for both the teacher and the student to have this kind of attitude, to try something new and to fail and try again, which is in itself already a challenge to bring that sort of mindset, also in the classroom.

**Speaker 4** [00:16:28] There seems to be an agreement also on the pedagogical side, to be active, to be hands on. So programming is a sort of a laboratory which also provides you with a number of interesting issues, because in order to have a laboratory, you have to have equipment. And also the assessment, one part seems to be in common, and there is this idea that the teacher should do formative assessment. You don't have to wait at the end, that you need to continuously be in touch with what students are doing. Because both you have to learn if what you are doing is working and you need to get the students on board and being confident of what they are learning. Do you have to provide them with some sort of the autonomy in assessing their own efforts as well.

**Speaker 2** [00:17:13] Looking a bit more at the future and what trends can be foreseen in the future across national curriculums when it comes to implementing or integrating computational thinking in compulsory education?

**Speaker 4** [00:17:25] Really looking at the future, would be interesting to start with the students that are now studying to be new teachers, but we also need to address some issues that we have at the moment. One has to do, how much room is going to be for this new subject. The problem is in schools you cannot just simply add, the time is always there, you have 5 hours and you can say, you can do another three other subjects, but the hours all always five. So what needs to be done and what needs not can be done? It's a complex issue and that's, practically, the need to reach an agreement. It will be nice if the ministry provides guidelines. One important part is the pedagogy, it's not just thinking about the teachers know now something, and if you need to let them know a little bit more about computer science, the real problem is what are the pedagogical problems, how you are going to teach those things? Computer sciences is a new subjects. And so, I think there probably there should be a effort in doing research on specifically on the subjects. And in the future, I think you know, probably going to be a little bit more clear guidelines and tools on how to asses.

**Speaker 5** [00:18:38] So, adding to what Augusto said, what we need to look at next and that is also something that is going to happen more and more in countries, is to look a bit into what is actually happening in the schools. So not only in the few best advanced schools, for example, that we talked to during our study. But what is happening in all schools in Europe, and we expect to find a lot of differences there, because all schools need the equipment, all schools need the teachers that are capable of teaching. And the strong assumption is that that at this very moment is not the case yet, but it's both for research to look more into that, but also countries themselves will now, since the curricula has been there for a few years, start monitoring and evaluating them. And also just to highlight, there are still a lot of open questions in research in policy making. So we still do not know exactly if what is taught now in curricula really fosters, these key competencies of the way curricula are set up now really prepare students for the future. And I personally think that only if we come to the conclusion that yes, and only if we know how to prepare teachers to teach that, only then we will have computational thinking and computer science with us for a very long time. And I also think it's you know, it's something we should discuss with everyone. So I would also invite all teachers for themselves to think critically. Why do I think my students need to learn computational thinking to really, you know, keep discussing these kind of questions to improve computer science education at the end of the day.

**Speaker 2** [00:20:08] Ind eed to believe in what is taught and to believe in the end, goal of what is taught and I think is a very good approach because of course, for the other subjects then it's been so for for many, many years. But however, with computational thinking, we have now a chance to implement it in a way that makes sense, in a way that is critical as well, that makes sense with the future of our students. And so I think that's a very good approach.

**Speaker 3** [00:20:29] I have a more concrete question for you. What are the key implementations and best practises that can be drawn from this study? What are the main findings here?

**Speaker 4** [00:20:40] If you want to sort of summarize, it's clear, then you need to upskill teachers. But the problem is not just upskill the content, but also to provide clear guidelines in the pedagogy. Also, you know, Ministeries, should be more clear on how much time, effort they want to dedicate to this new subject, not just add, but tell, you know, where the room is for addressing computational thinking and also provides clear strategies for the assessment, for the summative and the formative. If people, just see computational thinking this as the new Latin, you know, this big thing that is going to solve all the problems. It's not going to be. And also you know, we have been a little bit late. There has been a huge revolution in society. And we need to acknowledge it and all this digital technology also has a scientific side. It's good, too, the Ministry are taking care of their part, but it's not going to solve all the problems. So it needs to be taken with a grain of salt. And it is a long and difficult transformation, I think.

**Speaker 3** [00:21:50] Yes, indeed. That is being given more concrete now. What would you say, what a school should to do in order to better integrate computational thinking in their teaching practise?

**Speaker 5** [00:22:03] I would like to highlight again that it's not only the task of teachers and schools, but that it's really a common effort. And there are a lot of other actors that should provide support Ministries, other organisations, teacher-training organisations. So it's not something that schools should do themselves. And it's very likely that at this moment there will still be schools that will not find themselves in the ideal conditions, that do not have everything they need to implement computer science curriculum, as it should ideally be, but for schools themselves, first one thing we see also more and more is that school leadership is very important. One thing is to make time for the teacher to train themselves, to also help with finding relevant training opportunities, and also to really foster that culture of you can try something, you can innovate, it can go wrong, then we can discuss it. So we to encourage teachers to go in that way because of the responsibility is on the teacher alone, basically just do not figure it out. I wanted it to work and good luck, then it's a lot more difficult and for teachers themselves, I would encourage them to really look, who can help me, is that my colleague who already knows more? Or if in my school I happen to be the only computer science teacher, to who else can I reach out? There are so many online communities, there are school hubs where different schools connect. Go and find these opportunities and try it out. As we already said, formative assessment feedback is very important. Try something concrete. Ask your students. Don't be afraid to ask your students, how did it go? Was that interesting to you? Did you learn anything or wasn't it so useful? How can I improve it in the future? And take it from there. So it also, don't expect from yourself that you implement the perfect computer science curriculum immediately, try to find your way and try to also think how comfortable you can be with teaching something when you know there will be students in the room who know some things, better than, than you do.

**Speaker 3** [00:23:55] I would also like to point out that, how important it is to engage in professional networks and Code Week is such a network, our community of teachers from different types of schools really does a lot of different activities to help teachers integrate computational thinking in their classrooms.

**Speaker 2** [00:24:17] Indeed, I think it's it's important to remember what Arjana mentioned, about Code Week being also a space for reflection and the space for supporting, but also a space to to see the way that we teach coding and computational thinking. A critical view. For now, I want to bring it forward and to to look forward into these surveys, into these conclusions, right? So in all our podcast episodes, we like to end on a positive note and we like to end as well with some food for thought for our listeners. And so if you have to choose just one idea, our best practise from the study that should be implemented from now on, what would that be?

**Speaker 4** [00:24:51] I think, you know, this should be good move from top down and bottom up. Top down, I think would be nice to know, what I found interesting from the French case study, is they added computer science in the final examination. That's a clear indication, we are investing on it. Bottom up, that in the best my mind is creating a community of practises. In England that in the company, that school [00:25:15](?) [0.0s] it's it's a grassroots movement. So teachers. And now it's playing a big role, there is room for networking at the bottom level. That's probably the most interesting ideas that I found, in the end it works.

**Speaker 5** [00:25:29] For me, the main takeaway is also teaching communities obviously, and initiatives like the Code Week are very important. But I would if I could choose something that really goes even further, I would be even more specific. And as Augusto mentioned UK, England has a very interesting example. So, to really create school hubs that link schools, several schools in one region together, and also specific schools that are in more privileged areas and others that are in a more difficult situation. And also, provide them with funding, so that they really have also the time and the space to meet together. And then it becomes teachers that are more advanced training other teachers, but really also with the supports to properly do that and ideally even with universities on board. So then, it also becomes something that you can get research on board, but then you start doing research with schools, like really asking them, having them as partners on board, but not research about schools. So it's in the end really that idea of powering schools.

**Speaker 2** [00:26:30] Yeah. So in a way schools will become lighthouses rather than just part of the normal landscape. Yeah, I think I like a lot, those ideas that you mentioned and now to wrap up a bit, the episode. So we talked about this study and how it compares to the previous one and what new things we've learnt. I really like what you mentioned, that you found in general, a lot of similarities and I think that offers a lot of hope in terms of policy, if I can say so, in the sense that I think we have already some common strengths in Europe and some things that we can learn from each other. We talk a lot about teacher training, and I think that's something that we still need to look at and still need to work on. And we talked a lot about practises at school more particularly and how we teach computational thinking in different subjects and, and how this is still quite, quite different across countries, but I see through, through your answers that the teachers have also done a great deal to innovate in the way that they teach, and the way that they bring these topics to their students. And so I think that my, my main idea, after I listen to you both, after I read the study, I think there's a lot of work to be done. As we've discussed for sure, this quite novel topic, there's still quite a lot to be done, but I think there's quite, quite some, some future ahead, as quite already being done and having been done. So those are some of the highlights that I would nip pick, but Arjana, feel free to, to add.

**Speaker 3** [00:27:47] As you mentioned, integrating computational thinking is not an easy process. It's rather complex, but it is a topic very relevant for our students because these are the skills they need not only in the future, but also now when they are at school. And another thing that I would like to highlight, this teacher training, I would invite our listeners to check out materials and resources and guidelines that are already out there, so that they can use them in their classroom. And also, dear listeners, become part of a network and Code week is such a network, and we invite you to join us if you still haven't, and together with us, let's do more on the implementation of computational thinking in the classroom today.

**Speaker 2** [00:28:34] Katja, Augusto Is there anything you would like to add before we close the episode?

**Speaker 4** [00:28:37] Well, we want to thank you for this opportunity to have this exchange and for all the question and patience from your side, it was a nice opportunity. Thanks.

**Speaker 5** [00:28:47] I would just say one more thing. So I agree with you that as I said before, it was the only way of bringing something completely new to curricula to do it in the way it was done. So without having all teachers prepared for it, it was, in a way, a crazy adventure still. And overall, I guess teachers, as we know them, embraced that challenge. With a lot of innovation, with a lot of being brave to try out new things. But I also think it's okay if there are teachers now listening who are still not totally excited about computational thinking, that's fine. And you know, who are not... Have not dreamed about, you know, one day teaching computational thinking activities. I mean, look at the activities, look at what other teachers do and just look at what of these things make sense to them. But it is also okay not to be totally enthusiastic about computational thinking and then just go and have a look and see what of these things makes sense for.

**Speaker 3** [00:29:46] Katja and Augusto, thank you both for joining us for the EU Code week podcast!

**Speaker 3** [00:29:50] Thank you.

**Speaker 5** [00:29:51] Your welcome.

**Speaker 3** [00:30:02] We hope you enjoyed this episode and are now more aware of the importance of computational thinking in compulsory education, as well as both the obstacles and best practises to integrate it into the school curricula.

**Speaker 2** [00:30:16] And listeners, don't forget to tune in next week for a new episode of the EU Code Week podcast, when we'll discuss new insights about coding and technology and how we can integrate them into the classroom. But for now, goodbye. Goodbye.