European Commission

JRC SCIENCE FOR POLICY REPORT

# Developing Computational Thinking in Compulsory Education

*Implications for policy and practice*

**Authors**: Stefania Bocconi, Augusto Chioccariello, Giuliana Dettori, Anusca Ferrari, Katja Engelhardt

**Editors**: Panagiotis Kampylis, Yves Punie

2016

EUR 28295 EN

Joint Research Centre

**Contact information**
European Commission, Joint Research Centre
Address: Edificio Expo. c/Inca Garcilaso, 3. 41092 Seville (Spain)
E-mail: JRC-B4-SECRETARIAT@ec.europa.eu
Tel.: +34 95 44 88 401

**JRC Science Hub**
https://ec.europa.eu/jrc

JRC104188

EUR 28295 EN

PDF ISBN 978-92-79-64442-9 ISSN 1831-9424 doi:10.2791/792158

Luxembourg: Publications Office of the European Union, 2016



How to cite: Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). Developing computational thinking in compulsory education – Implications for policy and practice; EUR 28295 EN; doi:10.2791/792158



Abstract
**Developing Computational Thinking in Compulsory Education – Implications for policy and practice**

In the past decade, Computational Thinking (CT) and related concepts (e.g. coding, programing, algorithmic thinking) have received increasing attention in the educational field. This has given rise to a large amount of academic and grey literature, and also numerous public and private implementation initiatives. Despite this widespread interest, successful CT integration in compulsory education still faces unresolved issues and challenges. This report provides a comprehensive overview of CT skills for schoolchildren, encompassing recent research findings and initiatives at grassroots and policy levels. It also offers a better understanding of the core concepts and attributes of CT and its potential for compulsory education. The study adopts a mostly qualitative approach that comprises extensive desk research, a survey of Ministries of Education and semi-structured interviews, which provide insights from experts, practitioners and policy makers. The report discusses the most significant CT developments for compulsory education in Europe and provides a comprehensive synthesis of evidence, including implications for policy and practice.

# Table of contents

# Foreword

JRC research on Learning and Skills for the Digital Era started in 2005. The aim was to provide evidence-based policy support to the European Commission on harnessing the potential of digital technologies to encourage innovation in education and training practices; improve access to lifelong learning; and impart the new (digital) skills and competences needed for employment, personal development and social inclusion. More than 20 major studies have been undertaken on these issues resulting in more than 100 different publications.

Recent work on capacity building for the digital transformation of education and learning, and for the changing requirements for skills and competences has focussed on the development of digital competence frameworks for citizens (DigComp), educators (DigCompEdu), educational organisations (DigCompOrg) and consumers (DigCompConsumers). A framework for opening-up Higher Education Institutions (OpenEdu) was also published in 2016, along with a competence framework for entrepreneurship (EntreComp). Some of these frameworks are accompanied by (self-) assessment instruments. Additional research has been undertaken on Learning Analytics, MOOCs (MOOCKnowledge, MOOCs4inclusion) and policies for the integration and innovative use of digital technologies in education (DigEduPol).

This report on Computational Thinking (CT) aims to provide a comprehensive overview and analysis of recent research findings and grassroots and policy initiatives for developing CT as a competence for the 21st century among schoolchildren, and also to highlight the implications for policy and practice.

More information on all our studies can be found on the JRC Science hub: https://ec.europa.eu/jrc/en/research-topic/learning-and-skills.

**Yves Punie**

*Project Leader*
DG JRC Unit Human Capital and Employment
European Commission

# Acknowledgements

The authors and editors of this report would like to thank all the experts from the Ministries of Education who participated in the survey for their valuable support and additional information about relevant policies and documents in relation to the CompuThink study (see Annex 1).

Our special thanks go also to the experts, researchers, practitioners and policy makers who provided their valuable insights and knowledge on current developments in CT in primary and secondary education, highlighting the major implications for policy and practice. The list of experts interviewed in the CompuThink study is included in Annex 2.

We would also like to acknowledge and thank Jim Devine, Educational Consultant and former President of the IADT (IE), and Maciej Syslo, of the Universities of Toruń and Wrocław (PL), for providing valuable and detailed comments on earlier versions of this report.

Last but not least, thanks go to Jeffrey Earp, CNR-ITD, for reviewing and proof-reading the final version of this report and to Patricia Farrer, JRC, for proofreading the executive summary and conclusions.

# Executive summary

In recent years, Computational Thinking (CT) and related concepts (e.g. coding, programming, algorithmic thinking) have been promoted by educational stakeholders as skills that are as fundamental for all as numeracy and literacy.

A number of initiatives addressing CT and coding/programing have been carried out, both at international (e.g. EUCode week) and national levels (e.g. introducing programming into the statutory curriculum).

Despite the high levels of interest in developing CT skills among schoolchildren, however, a range of issues and challenges still needs to be addressed for the effective integration of CT in compulsory education. Key questions include: *How can we define CT as a key 21st century skill for schoolchildren?; What are the core characteristics of CT and its relationship with programming/coding in compulsory education?; How can teachers be trained to effectively integrate CT in their teaching practice?; Should CT be addressed within a specific subject (e.g. Computer Science, as part of STEM, or as a cross-curricular topic?; What does it mean to assess CT?; What is needed to further the CT agenda in compulsory education settings?*

## Policy context

In the context of the Digital Agenda, coding is explicitly regarded as a key 21st century skill: "*Coding is the literacy of today and it helps practice 21st century skills such as problem-solving, team work and analytical thinking*" (EU Digital Single Market, 2016). Along the same lines, the European e-Skills Manifesto (McCormack, 2014, p. 57) declares that *"...the world is going digital and so is the labour market... Skills like coding are the new literacy. Whether you want to be an engineer or a designer, a teacher, nurse or web entrepreneur, you'll need digital skills*."

In the Joint Report of the Council and the Commission on the Implementation of the Strategic Framework for European Cooperation in Education and Training – ET2020 (European Commission, 2015), the acquisition of digital competences, including coding, is regarded as essential to sustain economic development and competitiveness. Along the same lines, the New Skills Agenda (European Commission, 2016) explicitly invites Member States to develop "*coding / Computer Science*" in education. Innovation in education and training systems (E&T) and addressing digital skills are also highlighted in the Junker Commission's new strategy to deliver growth, jobs and investment in Europe (European Commission, 2016).

## Key conclusions

The evidence collected addresses each of the key questions outlined above and points to some recommendations on how the integration of CT in compulsory education might be tackled at policy-making level.

There is a lack of consensus on the definition of CT. Wing's (2011) landmark definition has become a reference point for discussion in the field, providing two valuable perspectives: (i) CT is a thought process, thus independent of technology; (ii) CT is a specific type of problem solving that entails distinct abilities, e.g. being able to design solutions that can be executed by a computer, human, or a combination of both. A few other definitions are present in the literature. Among the most cited, that by the Royal Society (2012) highlights the ability to recognise aspects of computation in the world that surround us.

In the absence of a single definition of this field, a set of core concepts and skills is again and again emerging from the literature to fill the gap. These include: *abstraction, algorithmic thinking, automation, decomposition, debugging, and generalization.* Coding/programming is a constituent of CT, in that it makes CT concepts concrete and can thus become a tool for learning, e.g. as a medium for exploring other domains or for self-expression (through the creation of digital storytelling and/or videogames). However, there is general consensus that CT actually entails much more than coding/programming. For example, the processes of *problem analysis* and *problem decomposition* precede coding/programming.

Several authors clearly distinguish CT from digital literacy/competence, pointing out that the distinctive characteristic of CT is its focus on problem-solving processes and methods, and on creating computable solutions. This is also reflected in the survey of Ministries of Education. While the two terms are clearly related, digital competence may not fully capture the set of core ideas and skills associated with CT.

Where CT is placed in school curricula varies from country to country. In some cases, it is integrated across subject areas, particularly at primary level, while in others it is part of a separate computing subject that is usually taught at secondary level. In addition, these two approaches are often combined. Some countries (e.g. Wales and Austria) consider CT and related concepts as part of the digital competence (DC) curriculum. This is also the case in the European Digital Competence Framework for Citizens.

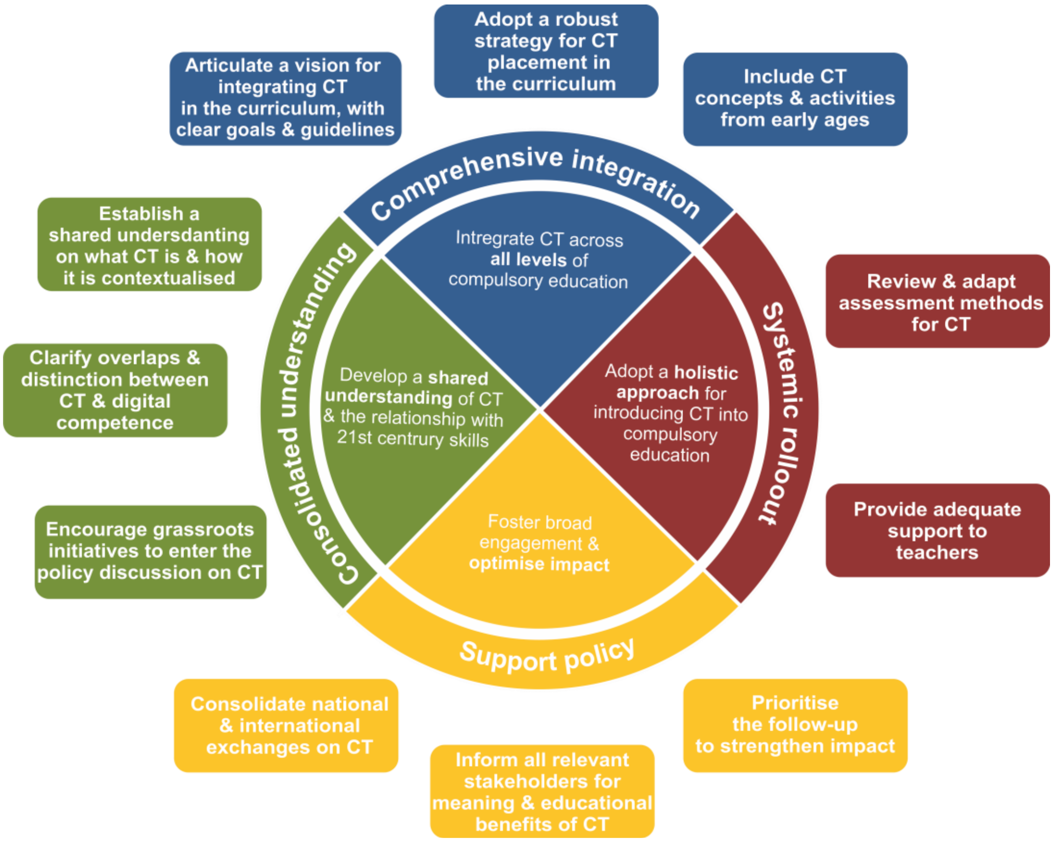
There is broad consensus among experts and practitioners that the introduction of CT in school curricula at all levels is creating demand for large-scale in-service continuous professional development. Training activities are often designed specifically to be hands-on so that teachers can more easily transfer their new skills to their classrooms. Grassroots efforts are also contributing to teachers' professional development.

Experts and practitioners are also emphasizing the importance of assessing students' ICT skills. However, only a limited amount of research has been carried out and currently there are only a few actual experiences of assessing students' grasp of CT concepts and of transferring of CT skills to other knowledge domains.

With regard to the integration of CT in compulsory education, four important areas emerge for policy makers and stakeholders to focus on: *consolidated CT understanding; comprehensive integration; systemic rollout;* and *policy support*. Recommendations for these areas are briefly summarized below.

Establishing a shared understanding of what CT is and how it is contextualized may facilitate the process of curriculum integration. At the same time, this solution will respect teachers' freedom to introduce CT in a way that is suited to their specific school context. This also entails clarifying the overlaps and distinctions between CT and digital literacy/competence. The experience gained and lessons learned from grassroots initiatives can also provide valuable input to the policy discussion.

In order for CT to be integrated comprehensively across all levels of compulsory education, it is necessary to define a clear vision and set specific goals. As CT involves far more than offering a few hours of coding, placing it in the curriculum calls for a robust strategy that takes into account the wide range of factors involved. A key consideration is the extent to which CT is allocated across the full spectrum of subject area studies and, also, in multi-disciplinary and inter-disciplinary contexts. Introducing CT concepts to children early on in school is commonly held to be desirable. These considerations call for a holistic approach to CT integration in compulsory education, which embraces essential aspects such as suitable assessment strategies and adequate teacher training.




Exchanges with multiple actors (policy makers, grassroots initiatives, research centres and other stakeholders) can yield extremely valuable insights and add value to policy actions. This is especially true at the pan-European level, where these exchanges can help avoid the repetition of mistakes and promote good practice. Another beneficial strategy for implementation is to ensure that ALL stakeholders, not just those directly involved in curricula development, are adequately informed about what CT is and how it is relevant to compulsory education. Finally, a wide-angle monitoring and analysis strategy is required to measure the impact and sustainability of implemented actions.

## Main findings

The study has gathered a wide-range of evidence from extensive desk research, a survey of Ministries of Education and interviews with experts.

A great variety of terms (e.g. coding, programming, algorithmic thinking) are used in the literature and in official policy documents to refer to CT. These reflect differing perspectives on CT (e.g. that it implies more than "computing"). In addition, each stakeholder may prefer to use different well-established terms (e.g. coding, problem-solving). Terminological variation also derives from the different contexts of use (documents for academic versus policy purposes) and from the way the terms are adopted in national languages.

The main rationale for introducing CT in most countries, in and outside Europe, is to foster the $21^{st}$ century skills necessary to fully participate in the digital world. In addition to pointing out the general benefits of CT as a thinking skill, many authors also stress the need to develop new skills for the employment market. The survey of Ministries of Education reveals a range of different reasons for integrating CT. Thirteen countries in Europe and beyond (AT, CH, CZ, DK, FI, FR, GR, HU, IT, LT, PL, PT, TR) aim to develop students' logical thinking skills and problem-solving skills through CT. Others, such as Finland and Portugal, also set quite specific goals, like raising student achievement and boosting interest in mathematics.

An upsurge in the integration of CT and, more broadly, of Computer Science in compulsory education is evident, as indicated by the recent wave of curricula reforms. Eleven countries in Europe and beyond (DK, FR, FI, HR, IT, MT, PL, TR, UK-EN, UK-SCT) have recently concluded a reform process that includes CT and related concepts. Seven others (CZ, GR, IE, NL, NO, SE, UK-WLS) are currently planning to introduce CT into compulsory education. Moreover, seven other countries (AT, PT, CY, IL, LT, HU, SK) are integrating CT by building on their long-standing tradition in Computer Science (CS) education, mainly in upper secondary schools. Some of these are expanding CS education to include the lower secondary and primary levels.

For those countries (namely: Spain, Germany, Belgium and Switzerland), where curricula development is managed at regional level, the integration of CT in school varies from region to region.

The study analysed how CT is positioned in the curriculum along two axes: educational levels and subjects. Most countries integrate CT in secondary school. However, there is a growing trend towards primary school integration as well. Several countries embed CT across subject areas, particularly at primary level, while at secondary level CT is mostly included as a computing subject in its own right. However, a combination of approaches is also present. Some countries (e.g. Wales and Austria) consider CT and related concepts as part of their digital competence (DC) curriculum.

### Related and future JRC work

CompuThink is an exploratory study that aims to contribute to the debate on coding, transversal skills, and competences at European and Member State level. It also links with the JRC studies on Digital Competence for citizens (DigComp), teachers (DigCompEdu) and schools (DigCompOrg).

### Quick guide

Computational Thinking (CT) is a thought process (or a human thinking skill) that uses analytic and algorithmic approaches to formulate, analyse and solve problems. In the past decade, CT has attracted increasing attention in the educational field, giving rise to a large amount of academic and grey literature, and also to numerous public and private implementation initiatives. Despite this widespread interest, successful CT integration in compulsory education still faces unresolved issues and challenges. The picture that emerges from this study shows a dynamic field. In Europe and beyond, the number of projects and experiences has been rapidly growing, along with increasingly widespread interest in understanding the nature of computational thinking and its contribution to 21$^{st}$ century skills.

The primary purposes of the CompuThink study are: (i) to provide a comprehensive overview of CT skills for schoolchildren, encompassing recent research findings and initiatives at grassroots and policy levels; (ii) gain better understanding of the core concepts and attributes of CT and its potential for compulsory education. The analysis carried out in the study focused on policy initiatives (in place or planned) that entail reform to national curriculum, and/or official guidelines, for integrating CT in compulsory education.

Numerous terms are currently being used that refer in some way to CT. Some of these emphasize a particular aspect of CT and/or reflect specific stakeholder positions. Moreover, context of use and national language also influence the terminology adopted.

The study adopts a mostly qualitative approach that comprises three main components: (i) extensive desk research that draws on a wide range of data sources, including academic and grey literature (e.g. journal papers, reports, blogs, etc.); (ii) a survey of Ministries of Education (MoEs) in Europe and beyond to gain access to documents of special relevance (e.g. curricula, guidelines) for further analysis; (iii) semi-structured interviews which elicit insights from experts, practitioners and policy makers. The key data gained from these sources were subsequently integrated into a coherent whole.

# 1. Introduction

Computational Thinking (CT) is the term in current use to refer to the key ideas and concepts of the disciplinary areas of Informatics and Computer Science (CS). In the past decade this topic has been gaining increasing attention from researchers, practitioners and policy makers in the education field. As a result, we are witnessing a significant increase in the amount of academic and grey literature on CT, which is also being mentioned, explicitly or implicitly, in policy-related documents. For instance, the New Skills Agenda for Europe (European Commission, 2016), although tackling digital skills for the general population rather than the formal education sector, explicitly invites Member States to develop CS in education: "*Member States, business and individuals need to rise to the challenge and invest more in digital skills formation (including coding / computer science) across the whole spectrum of education and training*."

A number of prominent institutions inside and outside Europe have intervened in the debate about the introduction of CT skills in compulsory education. In 2012, the Royal Society published the report "*Shut down or restart? The way forward for computing in UK schools*". The Académie des Sciences (2013) intervened on this subject in the report "*L'enseignement de l'informatique en France – Il est urgent de ne plus attendre*". Moreover, Informatics Europe and the Association of Computing Machinery (ACM) Europe, Working Group on Informatics Education (2013), urged Europe "not to miss the boat" on this subject. All those reports call for a change in curricula to make room for CS as a discipline. Industry also supports this position: following the Next Generation Report and Eric Schmidt's 2011 speech on the UK's education, the Department for Education decided to introduce CS teaching in primary and secondary state schools.

International debate has spotlighted the importance of CS studies not just as a crucial content area, but also for the potential to foster the development of general thinking skills and digital competences, especially regarding coding[1]. For instance, according to the Council of the European Union and the European Commission (2015), digital competence, one of the key competences for lifelong learning, encompasses coding: "*Learning and acquiring digital competences go beyond pure ICT skills and involve the safe, collaborative and creative use of ICT, including coding*". The research work discussed in this report seeks to contribute to the current debate on CT, coding/programming and transversal skills at European and global level. Particular attention is focused on the following aspects:

- How can we define CT as a key 21st century competence for schoolchildren?
- What are the core concepts and skills of CT?
- What is the relationship with programming/coding in compulsory education?
- What is the relationship with digital literacy/digital competence?
- How should teachers be trained to effectively integrate CT in their teaching practice?
- Should CT be addressed within a specific subject (e.g., CS), integrated in STEM, or treated as a cross-curriculum topic?

[1] A variety of terms are used with reference to Computation Thinking depending on multiple factors and contexts. For an overview of CT-related terms see Box 1.

- What does it mean to assess CT?
- What is needed to further the CT agenda in compulsory education settings?

CompuThink was designed and funded by the European Commission's Joint Research Centre (JRC), Directorate B – Innovation and Growth, Human Capital and Employment Unit. The study was launched in December 2015 and carried out by the Institute for Educational Technology of the Italian National Research Council, together with European Schoolnet.

# 2. Objectives and methodology

This study aims to provide a comprehensive overview of CT skills for schoolchildren, encompassing recent research findings and initiatives at grassroots and policy levels. The ultimate goal is to better understand the core concepts and attributes of CT as a key competence for schoolchildren, and its potential for compulsory education. The specific objectives of CompuThink are to:

- thoroughly and comprehensively review definitions and frameworks of CT skills in the context of compulsory education (i.e. primary, secondary and initial VET schools);
- analyse and synthesise findings generated by grassroots and policy initiatives for the development of CT skills in schoolchildren, inside and outside formal education settings;
- document the developments in the CT field that are most significant for compulsory education in Europe and provide a comprehensive synthesis of evidence, including implications for policy and practice.

The study follows a mostly qualitative approach that comprises four main components:

- an extensive desk research that draws on a wide range of data sources, including both academic and grey literature (e.g. journal papers, reports, blogs, etc.);
- survey of Ministries of Education (MOEs) in Europe and beyond in order to gain access to documents of special relevance (e.g. curricula, guidelines) for further analysis;
- semi-structured interviews providing insights from experts, practitioners and policy makers;
- integration of key data into a coherent whole.

These are described in more detail in the following sections.

Figure 1 depicts the overall methodological approach and main components of the study.

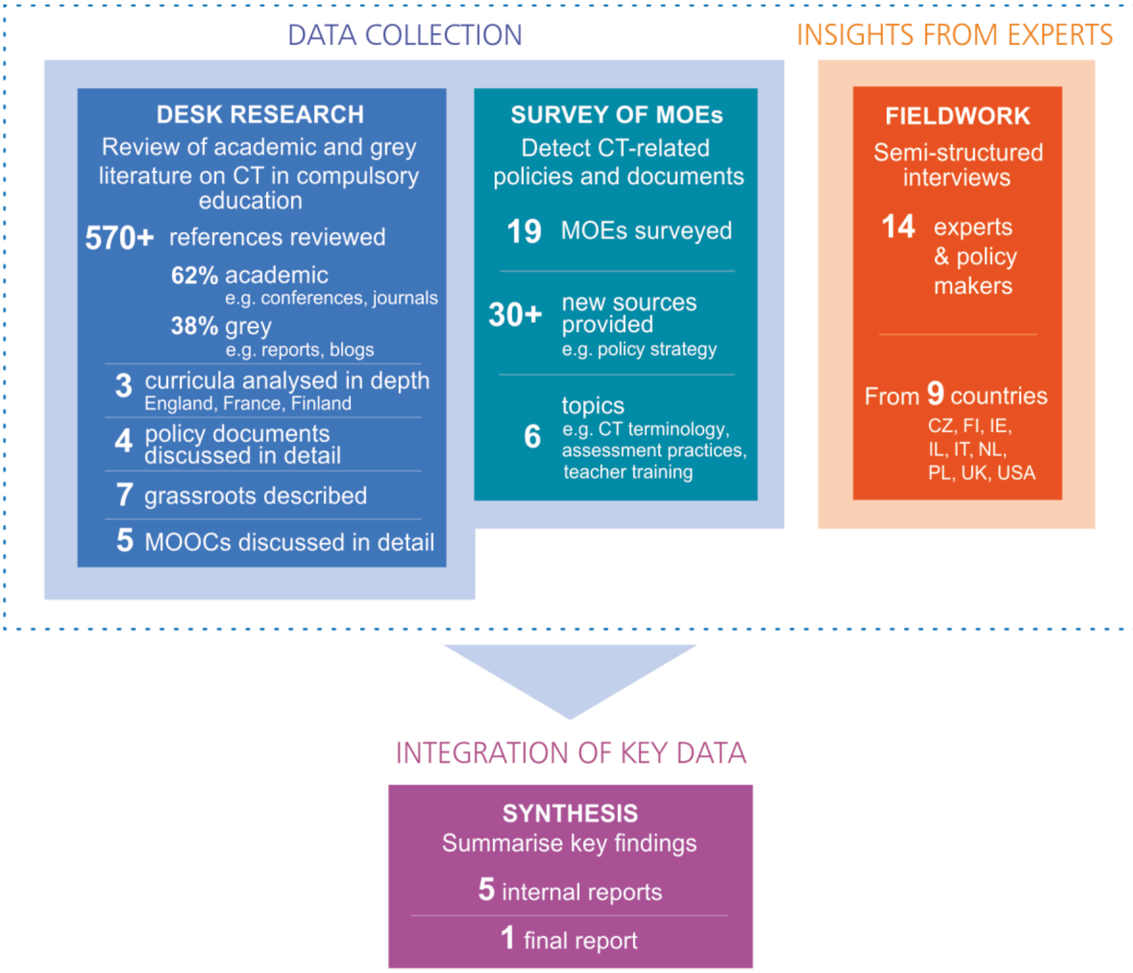



**Figure 1.** Overall approach of the CompuThink study

## 2.1 Desk research

An extensive review of the literature was carried out using a wide range of sources, such as technical and policy reports, journal and conference papers, book chapters, websites and blogs, press clips and newspapers, video clips, and presentations. The review had a two-fold objective:

- to define and classify existing conceptualizations of CT as a key 21st century skill (or set of skills) in the context of compulsory education;
- to collect evidence of policy initiatives and practical implementations, including grassroots initiatives and MOOCS for developing CT skills among primary/secondary students.

A review matrix outlining the main research studies, findings and implications was used to structure the in-depth and comparative analysis of the sources and to identify key issues. Inter-rater reliability ensured that the various researchers involved in the project analysed and tagged selected items in a homogeneous and comparable way. The review matrix touched upon a number of the study's research questions, including the definition of CT and CT skills (or set of skills), and also its relation to programming /coding and to digital literacy/digital competence.

The wide-angle literature search gathered more than 570 sources from both academic and grey literature. Moreover, three recently updated national core curricula (from England, France and Finland) were analysed in-depth. In addition, four European-level policy documents were discussed, along with seven grassroots CT initiatives implemented both inside and outside Europe. More than twenty Massive Online Open Courses (MOOCs) related to CT were described in detail.

## 2.2 Survey of MOEs

The desk research was complemented by a survey of various Ministries of Education in Europe plus Israel and Turkey[2]. This survey was instrumental for identifying and gaining access to more than 30 documents of particular relevance for CT, such as curricula, guidelines, policy strategies and country reports. These focused documents provided additional information on implementation plans (i.e. programmes of study) and/or supporting measures (i.e. teacher training and assessment). Specifically, the survey covered six major CT topic areas, namely: terminology, curriculum integration, preparation of teachers, assessment strategies, relationship with digital literacy/digital competence and relationship with coding/programming. The documents identified in the survey were integrated into the desk research, while the insights gained into current ministerial priorities and work in progress supplemented the findings from the desk research and the expert interviews.

## 2.3 Insights from experts

Policy makers, researchers and practitioners were consulted to collect informed insights via semi-structured interviews. The key aim of the interviews was to validate and complement the desk research. The fourteen separate interviews involved experts and policy makers from nine countries. The list of all interviewed experts is provided in Annex 2.

[2] The Members of the EUN Steering Committee (31 Ministries of Education) were invited to fill in the survey. 18 countries replied, namely: Austria, Czech Republic, Denmark, Estonia, Finland, France, Hungary, Israel, Italy, Lithuania, Malta, Norway, Poland, Portugal, Spain, Sweden, Switzerland, Turkey. The Greek MOE did not reply to the survey questionnaire. However, the Greek Institute of Education Policy and the MOE Directorates for primary, secondary and VET education provided the CompuThink team with general information on the status of CT in the Greek education system.

# 3. Understanding Computational Thinking

## 3.1 CT definitions

Jeanette Wing introduced the term 'computational thinking' in a viewpoint column published in Communications of the ACM in March 2006 (p. 33):

> *"Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science."*

Wing claimed that CT "*represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use*" (p. 33). This article has since stimulated a lively international debate on the nature of CT and its value to education, with contributions from academia, education, industry, and policy makers.

In 2010, the US National Research Council (NRC) organized a "Workshop on the Scope and Nature of CT" with key international researchers, including Jeannette Wing. One outcome was the evident lack of consensus on basic definitions, as participants expressed differing views about the scope and nature of CT.

In 2011, to move the discussion forward, Jeannette Wing proposed a new definition of CT:

> *"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 effectively carried out by an information-processing agent"* (p. 1).

Two aspects emerge from this definition that are particularly significant for compulsory education:

1. CT is a thought process, thus independent from technology;
2. CT is a specific type of problem-solving that entails distinct abilities, e.g. being able to design solutions that can be executed by a computer, a human, or a combination of both.

Wing's definition has subsequently become a reference point for discussion on CT. This notwithstanding, other distinct definitions have emerged in the literature. Among the most cited of these is the definition the Royal Society proposed in 2012 (p. 29), which stresses that computation is not exclusively a human construct but is also present in nature, for example in DNA:

> *"Computational thinking is the process of recognising aspects of computation in the world that surrounds us, and applying tools and techniques from Computer Science to understand and reason about both natural and artificial systems and processes"*.

The Computer Science Teachers Association and the International Society for Technology in Education (CSTA & ISTE, 2009, p. 1) have developed an operational definition that serves as another significant reference point. This lists all the operations that constitute CT as a practice:

> *Computational Thinking (CT) is a problem-solving process that includes (but is not limited to) the following characteristics*:
>
> - *Formulating problems in a way that enables us to use a computer and other tools to help solve them;*
> - *Logically organizing and analysing data;*
> - *Representing data through abstractions such as models and simulations;*
> - *Automating solutions through algorithmic thinking (a series of ordered steps);*
> - *Identifying, analysing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources;*
> - *Generalizing and transferring this problem-solving process to a wide variety of problems.*

In August 2016, the CSTA released the [Interim] CSTA K–12 Computer Science Standards. This update to existing CSTA standards refers to Wing's (2011) CT definitions and stresses the problem-solving aspects, as well as abstraction, automation, and analysis as distinctive elements of CT:

> *"We believe that computational thinking is a problem-solving methodology that expands the realm of computer science into all disciplines, providing a distinct means of analysing and developing solutions to problems that can be solved computationally. With its focus on abstraction, automation, and analysis, CT is a core element of the broader discipline of computer science"* (p. 6).

## 3.2 Core CT concepts and skills

Numerous papers examine the potential advantages of introducing CT in compulsory education. The belief is that CT can enable children and young people to think in a different way while solving problems, to analyse everyday issues from a different perspective (Lee et al., 2011), to develop the capacity to discover, create and innovate (Allan et al., 2010), or to understand what technology has to offer.

Kolodner believes that CT is a set of skills that transfers across disciplinary domains (NRC 2011, p. 54). In Resnick's view, CT is not simply a way to learn problem-solving skills but also a means for expressing oneself with digital media. This means that CT capacities are needed for design and social cooperation (ibid, p. 68).

Different authors suggest a wide variety of skills related to CT acquisition, such as: problem-solving, examining data patterns and questioning evidence (Charlton & Luckin, 2012); collecting, analysing and representing data, decomposing problems, using algorithms and procedures, making simulations (e.g Gretter & Yadav, 2016); using computer models to simulate scenarios (Creative Learning Exchange, 2015); dealing with open-ended problems and persisting in challenging cases (Weintrop et al., 2015); and reasoning about abstract objects (Armoni, 2010).

This variety also emerges from the interviews with experts. Irene Lee emphasizes that humans are formulating problems and designing solutions to be carried out by computers; hence, CT entails mapping solution processes onto computer capabilities, such as iteration, selection, and sequencing. In Tullia Urschitz's view, CT involves breaking a problem into smaller components, finding solutions (algorithms), writing instructions, and analysing one's own solution. According to Joke Voogt, CT is closely connected to Computer Science, especially in relation to characteristics like abstraction, problem decomposition and automation. Jan Lepeltak emphasises the strong connection between CT and language, namely that CT concerns not only how computers work, but also how we can communicate with computers and talk about information technology.

Mitchel Resnick also stresses the connection with language, viewing computation as literacy: CT is a way of expressing ourselves and understanding the world using computers and computational ideas. Two interviewees, Judith Gal-Ezer and Leo Pahkin, prefer to use the expression "algorithmic thinking" rather than "computational thinking". Finally, Maciej M. Sysło pointed out that there is no agreement on the definition of the term, also because the concept is still in its infancy. Janusz Krupa expressed concern about the consequences of introducing the term CT. Even though the new Polish curriculum refers to CT, Krupa reports that teachers might be "*afraid of CT*", hence it would be better to use more familiar expressions like problem solving, algorithmic thinking and critical thinking.

In spite of the wide variety of definitions and proposals, a subset of core concepts and skills is recursively emerging from the literature. In Table 1 we juxtapose CT skills identified in five prominent papers, which were selected for: a) being highly cited; b) having wide scope in reporting other studies; c) providing a variety of perspectives and points of view in terms of research strands and international working groups (e.g. the CSTA task force on CT, Computing At School, IFIP's Task Force Curriculum and the EDUsummIT TWG9 Curriculum group). The skills proposed in these seminal papers are, moreover, in line with skills emerging from the rest of the literature reviewed for the CompuThink study.

**Table 1.** CT concepts and skills in the literature

| Barr & Stephenson, 2011 | Lee et al., 2011 | Grover & Pea, 2013 | Selby & Woollard, 2013 | Angeli et al., 2016 |
|---|---|---|---|---|
| Abstraction | Abstraction | Abstractions and pattern generalizations | Abstraction | Abstraction |
| Algorithms & procedures | | Algorithmic notions of flow of control | Algorithmic thinking | Algorithms (including Sequencing and Flow of control) |
| Automation | Automation | | | |
| | Analysis | | | |
| | | Conditional logic | | |
| Problem Decomposition | | Structured problem decomposition (modularizing) | Decomposition | Decomposition |
| | | Debugging and systematic error detection | | Debugging |
| | | Efficiency and performance constraints | Evaluation | |
| | | | Generalizations | Generalization |
| | | Iterative, recursive, and parallel thinking | | |
| Parallelization | | | | |
| Simulation | | | | |
| | | Symbol systems and representations | | |
| | | Systematic processing of information | | |

Several of the above items are also present in Wing's articles on CT:

- The most important and high-level thought process in CT is the **abstraction** process (2011, p. 1);
- An **algorithm** is an abstraction of a process that takes inputs, executes a sequence of steps, and produces outputs to satisfy a desired goal (2011, p. 1);
- Computing is the **automation** of our abstractions. We operate by mechanizing our abstractions, abstraction layers and their relationships. Mechanization is possible due to our precise and exacting notations and models (2008, p. 3718);
- CT is using abstraction and **decomposition** when attacking a large complex task or designing a large complex system (2006, p. 33);
- Abstraction is used in defining patterns, generalizing from instances, and parameterization [**generalization**] (2011, p. 1).

The terms gleaned from the articles considered in Table 1 and from Jeannette Wing's papers point to a clear characterisation of CT. We might say that CT describes the thought processes entailed in formulating a problem so as to admit a computational solution involving abstraction, algorithmic thinking, automation, decomposition, debugging and generalization. The definitions of these items are provided in Table 2.

**Table 2.** CT core skills and definitions

| CT Skill | Definition |
|---|---|
| **Abstraction** | Abstraction is the process of making an artefact more understandable through reducing the unnecessary detail. The skill in abstraction is in choosing the right detail to hide so that the problem becomes easier, without losing anything that is important. A key part of it is in choosing a good representation of a system. Different representations make different things easy to do (Csizmadia et al., 2015, p. 7). |
| **Algorithmic thinking** | Algorithmic thinking is a way of getting to a solution through a clear definition of the steps (Csizmadia et al., 2015, p. 7). |
| **Automation** | Automation is a labour saving process in which a computer is instructed to execute a set of repetitive tasks quickly and efficiently compared to the processing power of a human. In this light, computer programs are "automations of abstractions" (Lee, 2011, p. 33). |
| **Decomposition** | Decomposition is a way of thinking about artefacts in terms of their component parts. The parts can then be understood, solved, developed and evaluated separately. This makes complex problems easier to solve, novel situations better understood and large systems easier to design (Csizmadia et al., 2015, p. 8). |
| **Debugging** | Debugging is the systematic application of analysis and evaluation using skills such as testing, tracing, and logical thinking to predict and verify outcomes (Csizmadia et al., 2015, p. 9). |
| **Generalization** | Generalization is associated with identifying patterns, similarities and connections, and exploiting those features. It is a way of quickly solving new problems based on previous solutions to problems, and building on prior experience. Asking questions such as "*Is this similar to a problem I've already solved?*" and "*How is it different?*" are important here, as is the process of recognising patterns both in the data being used and the processes/strategies being used. Algorithms that solve some specific problems can be adapted to solve a whole class of similar problems (Csizmadia et al., 2015, p. 8). |

Some authors claim that CT is not only characterised by skills, but also by attitudes or dispositions, as outlined in Table 3 below.

**Table 3.** CT dispositions / attitudes / attributes

| Reference | CT dispositions / attitudes / attributes |
|---|---|
| **Barr, Harrison & Conery** (2011, p. 51) | Confidence in dealing with complexity<br>Persistence in working with difficult problems<br>The ability to handle ambiguity<br>The ability to deal with open-ended problems<br>The ability to communicate and work with others to achieve a common goal or solution |
| **Woollard** (2016, p. 5) | Tinkering<br>Creating<br>Debugging<br>Persevering<br>Collaborating |
| **Weintrop et al.** (2015, p. 133) | Confidence in dealing with complexity<br>Persistence in working through challenging problems<br>Ability to deal with open-ended problems |

If we consider a competence as being the sum of knowledge, skills and attitudes, the key points summarized in Table 3 raise the possibility of considering CT as a competence. In her interview, Voogt emphasised that:

> "*we know from research that an important attitude for CT, which goes with this set of skills, is that students are able to work with uncertainty in complex situations, as well as having to be precise. Hence, there certainly are a number of attitudes that are also being developed while developing CT skills; for this reason, speaking of CT as a competence is reasonable*."

## 3.3 CT's relationship with Digital Competence

In 2006, the European Parliament and the Council of the European Union published recommendations on key competences for lifelong learning. Digital competence is one of the eight competences identified in the document (European Parliament, 2006). In the same year, the term computational thinking was proposed by Jeanette Wing as shorthand for thinking like a computer scientist, thus focusing attention not on the confident use of technology but on understanding its underlying core concepts. The current trend of integrating CT in compulsory education makes the exploration of its relationship with the term digital competence even more relevant.

The desk research identified only a small number of references that explicitly addressed this aspect; by comparison, more references were found to the terms digital literacy, ICT literacy and technology literacy[3]. This can be explained by variations in vocabulary: as a term, digital competence is mainly used by the European Commission and by Norwegian scholars, while in the Anglo-Saxon world there is a tendency to speak about digital skills or digital literacy. In broad reflection about introducing CT skills in compulsory education, Yadav (2014) points out how CT moves students beyond operational and technical skills, creating problem solvers instead of software users, encouraging creativity and problem-

[3] Discussing the differences and overlapping of these terms is beyond the scope of this report.

solving and enhancing many of the problem-solving techniques teachers already know and teach. Zapata-Ros (2015) also advocates CT as a competence *per se*, entailing 14 interconnected elements that include, among others, creativity and metacognition, together with abstraction and recursion. Gretter and Yadav (2016) present two approaches to 21st century skills that merge CT with UNESCO's concept of Media & Information Literacy (MIL) in support of students' 21st century skills and citizenship. They also discuss how CT and MIL together can equip students with the complementary skills to become active as well as reflective participants in their digital culture. They conclude that "*the complementary relationship between computational thinking and media & information literacy can provide teachers with a comprehensive set of skills to allow students to both critically navigate and creatively produce digital content*" (p.6).

This resonates with the view Resnick voiced in the interviews about computing being a medium for self-expression: "*programming, like writing, is a means of expression and an entry point for developing new ways of thinking*". It should also be noted that the European reference framework for the digital competence of citizens, DigComp (Ferrari, 2013), includes programming. The recent update, DigComp 2.0 (Vuorikari et al., 2016), encompasses the main components of Information Literacy and parts of UNESCO's Media & Information Literacy.

A strong consensus emerges from the literature that CT is more than programming, and the relationship with digital competence might not be able to capture fully the core ideas and skills associated with CT. Voogt points out that digital literacy differs somewhat from CT because it concerns basic literacy skills. Conversely, if everyone acquired some fundamentals of CT, they would gain a better understanding of technological development, helping them to master technological development instead of fearing it.

The most significant contributions on the relationship between digital competence and CT come from policy documents and from discussions on the implementation of CT and CS in school education. CT is also associated with an implicit criticism of the ways that digital literacy has been addressed in practice at schools. Consequently, there is a push to concentrate not on the technology but on the ideas and the science behind the technology of the digital revolution. A number of influential papers clearly expressed this idea:

- *Running on Empty: The Failure to Teach K–12 Computer Science in the Digital Age* (ACM & CSTA, 2010)
- *Shut down or restart? The way forward for computing in UK schools* (The Royal Society, 2012)
- *L'enseignement de l'informatique en France - Il est urgent de ne plus attendre* (Académie des Sciences, 2013)
- *Informatics education: Europe cannot afford to miss the boat* (Informatics Europe & ACM Europe Working Group on Informatics Education, 2013).

All these documents combine advocacy for including CT in school Computer Science studies with a critique of the way digital literacy is dealt with in education. For example, Simon Peyton Jones reports that in England "*there was a statutory subject called Information and Communication Technology (ICT); however, it was a technological subject focused on how to use artefacts. A variety of factors made ICT a low status subject especially in the eyes of students*." This resulted in a complete restructuring of the ICT syllabus, now called *Computing*. Hence, computer science is introduced along with digital and information literacy, as advocated by the Royal Society report (2012, p. 11): "*The term ICT as a brand should be reviewed and the possibility considered of disaggregating this into clearly defined areas such as digital literacy, Information Technology and Computer Science. […] The term 'ICT' should no longer be used as it has attracted too many negative connotations*."

Results from the survey of ministries reflect similar discussions on the relationship between CT and digital competence/digital literacy. Norway puts forward a vision that emphasis CT as a means to understand what lies 'behind the curtain', and how those tools actually work. This conceptualisation clearly distinguishes CT from digital literacy, which tends to focus on being a competent and safe user of digital tools and resources. Another distinctive element of CT that Norway puts forward is a focus on problem-solving processes and methods, and on creating solutions. According to the Czech Republic, CT is regarded more as a competence, while a well-developed digital literacy is a precondition for thinking in a "computational" way. According to Hungary, by observing and understanding the algorithms of the functioning computer, an aptitude towards computer usage can be developed. In Italy, CT is seen as key to digital and media literacy, an indispensable alphabet for student awareness of the digital environment and the capacity to proactively create and act in the digital world. According to Lithuania, developing CT can help to develop digital skills and collective intelligence. A similar view is expressed in Poland, since the new CS curriculum addresses all students in compulsory education and, it is held, will also contribute to general digital literacy. In the Maltese primary cross curriculum, CT is ingrained in the subject of Digital Literacy and ICT. Finally, Wales has included CT in their Digital Competence Framework (DCF) adopted in September 2016.

## 3.4 CT's relationship with coding and programming

Coding and programming are often used interchangeably to indicate the process of 'writing' instructions for a computer to execute. However, programming refers to the broader activity of analysing a problem, designing a solution and implementing it. Coding is the stage of implementing solutions in a particular programming language. Implementation skills go beyond coding since they include debugging and testing (Duncan et al., 2014).

In general, it is agreed that CT and programming are not overlapping sets: "*thinking as a computer scientist means more than being able to program a computer*" (Wing, 2006 – p. 33). Voogt pointed out in the interviews that while coding and programming are an important part of CT, CT certainly entails other core elements such as problem analysis and problem decomposition.

Simon Peyton Jones expressed the view that CT is the ability to: a) develop computational abstractions of real-world problems; and b) design, develop, refine and reason about computation artefacts (i.e. programs). The English Computing Curriculum states that even primary children should be able to "*use logical reasoning to predict the behaviour of simple programs*"; that is, being able to explain to somebody else what a program is intended to do or, if a program is not behaving as expected, understand why. Predicting is important: programming is not just writing but also being able to execute mentally what is written. This is core to CT.

Despite these distinctions, programming can make CT concepts concrete and become a tool for learning. Several authors instantiate the role of programming in the context of a CS curriculum; others see programming as a medium to explore other domains or for self-expression through the creation of digital storytelling and/or videogames.

Since abstraction is a key CT skill, some authors think that the introduction of CT & programming at the primary school level calls for empirical research on the earliest age in which students can handle abstraction, and specifically do abstraction (Armoni & Gal-Ezer, 2014). Others conceptualize CT as more aligned with language and literacy, especially the writing part of literacy. In their view, programming is a form of writing to express oneself with computational media.

Mitchel Resnick sees writing and programming as being abstract in similar ways: "*Writing words is very abstract, what's the meaning of those lines on the paper. Five-year-old children know how to put things into a sequence; if the programming environment enables [one] to easily put commands into a sequence to make interesting things happen, children will do it*".

**Box 1.** CT-related terms

The analysis of collected sources reveals that a variety of terms are used with reference to CT. Figure 2 depicts the set of CT-related terms considered in this study in order to capture major trends in the field.

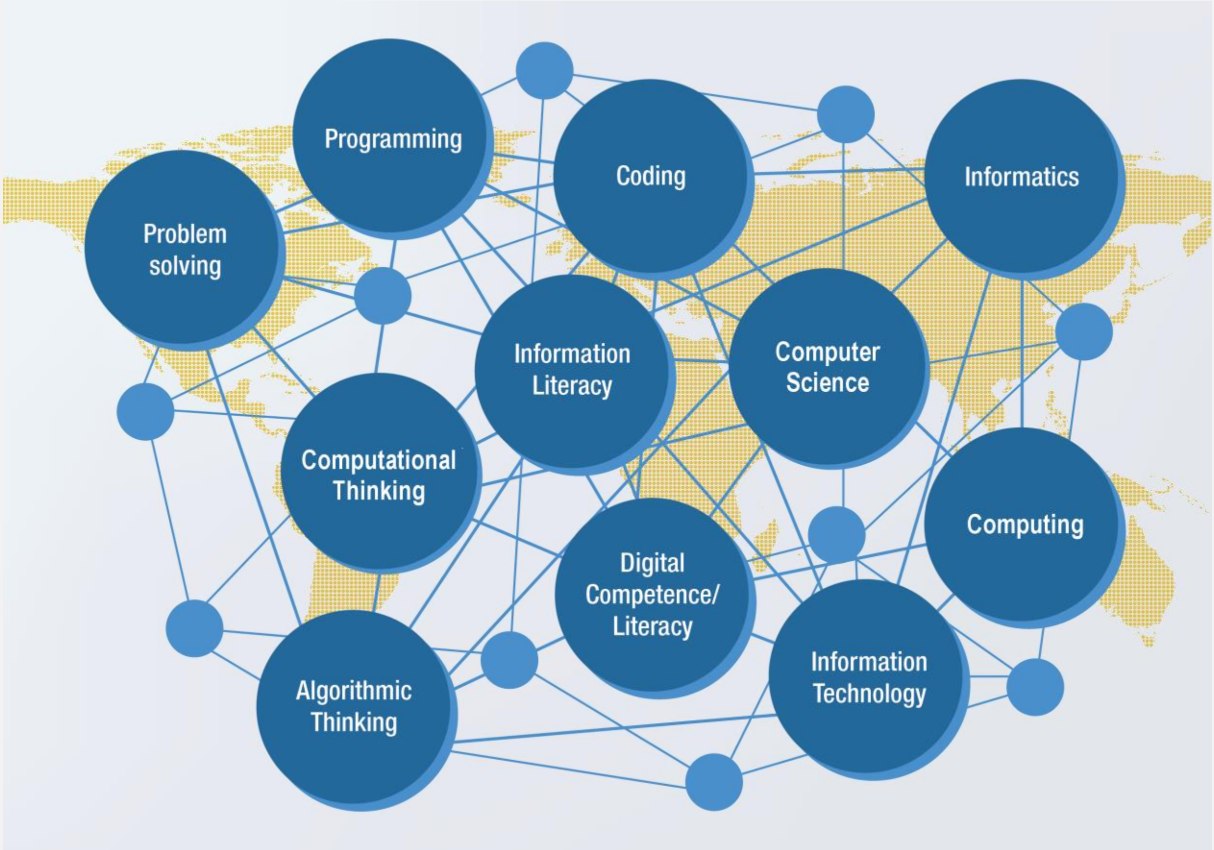



**Figure 2.** Set of CT-related terms addressed in this study

Some specific terms used in relation to CT emphasize a particular aspect and/or reflect stakeholders' positions:

> *"I prefer to use the expression Algorithmic Thinking (AT) for the kind of skills that people include in CT, because "computational" has to do with computing, while in this field we do much more than computing, we solve problems by going through several levels of abstraction; at the end, we also have a computation but this is just a final stage. AT is the spirit of computing, the art of computing."* (Judith Gal-Ezer, Israel)

> *"Teachers are rather afraid of the term CT. Terms like problem-solving, algorithmic thinking and critical thinking are better known. These terms are easier to understand and mean almost the same – the only difference being the use of ICT tools."* (Januz Krupa, Poland)

The context of use (academia versus policy documents) also influences the terminology adopted in referring to CT:

> *"The term CT is the most appropriate to use, albeit a bit technical. The Maltese National Curriculum does not refer to CT, as this term is reserved for academic research and official documentation."* (Maltese respondent to the survey)

The relative firmness of the term CT in national languages also impacts on the choice of term used:

> *"It is possible to translate CT into Finnish, but the Finnish term sounds too much, as if humans started to think like a machine and then we are not humans anymore. Therefore, it is better to use a more general term"* (Leo Pahkin, Finland)

The ministerial survey confirmed the points made by the Finnish expert. Figure 3 below shows MOEs responses to the survey's question: "*Which term do you use in your own national language to refer to CT?" What is the corresponding term in English?*"

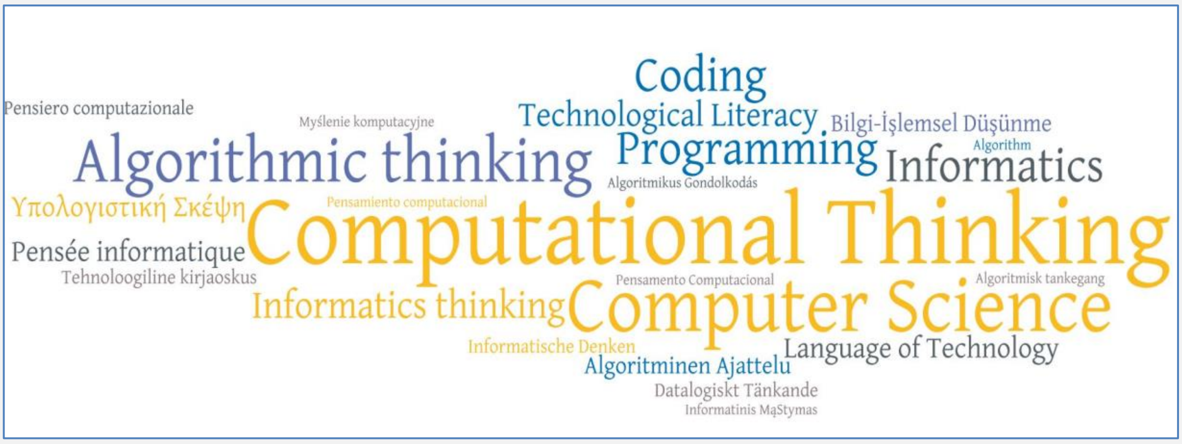



**Figure 3.** The term "computational thinking" in different languages

# 4. Major trends in CT integration within compulsory education

Initiatives to include CT concepts in compulsory education curricula are being undertaken in various parts of the world. In this section, we provide an overview of the current situation in Europe plus Israel and Turkey, as revealed by the information collected for this study [4]. This was derived from the survey of MOEs, together with the expert interviews and desk research.

Curriculum is an overloaded term that refers both to a *policy-related curriculum* and to a *conceptual-pedagogical curriculum*. In this study we focus on policy initiatives (in place or planned) that entail reform to national curricula, and/or to official guidelines, in which integration of CT in compulsory education is included; national initiatives that do not reflect these criteria were excluded. One such exclusion emerging from the survey of MOEs was Estonia. Contrary to what is usually reported in the press, Estonia actually has no policy or current plans for including CT in the national curriculum. The main CT-related initiative, ProgeTiger, supports Estonian schools willing to implement coding and CT, robotics and 3D technology at local level. It was launched in 2012 and is currently coordinated by Estonia's Information Technology Foundation for Education (HITSA) in conjunction with the Estonian Ministry of Education and Research. While 80% of Estonian schools have benefited from the ProgeTiger programme, not all schools are involved or participate. ProgeTiger is currently conducting a survey to analyse ICT and technology usage in school curricula and this is scheduled to conclude in March 2017.

## 4.1 Rationale for including CT in curricula and official guidelines

A number of articles included in our desk research address the rationale for including CT in the current provision for formal education. Many authors see CT in curricula as a means for developing learners' problem-solving abilities. Webb and colleagues (2016) argue that aspects of Computer Science, including programming, provide an ideal way of developing computational thinking, which learners can then apply more broadly as a problem-solving strategy. In the curriculum for Informatics in Austria, understanding of Informatics is seen as a way of enabling problem solving: through analysis of real processes in their personal environment, students should become capable of understanding complex systems and interdependencies [1].

CT and related concepts are seen as a form of empowerment. The French working group of the National Digital Council [20] maintains the entitlement of students to be active digital citizens capable of leading the digital transformation, rather than being subject to it. The Australian curriculum considers it important for students to learn how to use and develop digital technologies in order to participate fully in the digital world [70].

Introducing CT is also seen as a way of bridging the gap between curricula and the current needs of learners and society in general. A major concern in the UK is that the

4 For each country, key sources of information reviewed in the study are listed in Annex 3. Those references are cited in the text using a progressive number (e.g. [1]).

old ICT curriculum had become unbalanced, with too much emphasis on so-called basic digital skills at the expense of deeper understanding of concepts (Webb et al., 2015).

The economic recovery rationale seems to be the main driver underlying the *CS for all* initiative recently initiated by US President Barak Obama. This aims to empower all American students, from kindergarten to high school, by giving them the computer science skills they need to thrive in a digital economy. The economic argument can also be seen in the expectancy that young people exposed to CT and CS at school will choose to pursue CS studies in the future.

Summarising, two main trends emerge regarding the rationale for including CT in compulsory education:

1. developing CT skills in children and young people to enable them to think in a different way, express themselves through a variety of media, solve real-world problems, and analyse everyday issues from a different perspective;
2. fostering CT to boost economic growth, fill job vacancies in ICT, and prepare for future employment.

Out of the countries that answered this survey question, 13 (AT, CH, CZ, DK, FI, FR, GR[5], HU, IT, LT, PL, PT, TR) aim to develop students' logical thinking skills. All these countries also aim to develop problem-solving skills, while 11 (all except for DK and HU) aim to foster other key competences. This last category includes, in the case of Lithuania, fostering students' competence to organize and analyse data. In Finland, CT also supports other transversal competences in the national core curriculum, such as learning to learn, cultural competence, multi-literacy, ICT competence and entrepreneurship. Moreover, Finland aims to strengthen students' motivation to study mathematics.

Seven of the 13 countries (FI, FR, LT, PL, PT, CH, TR) focus on the development of coding and programming skills in particular. Attracting more students to study computer science is also a rationale for five of these (FI, FR, LT, PL, TR); fostering employability in the ICT sector is an aim for only three of the countries (FI, FR, TR) (see Table 4). Several European countries (CZ, FI, PL, NL) view development of CT skills as a way to prepare students for life in the digital world, a point also stressed in the expert interviews. One argument put forward is that school education needs to follow societal developments. The Finnish expert suggested that everybody should have a basic understanding of how the machines that surround us work. Moreover, the Polish expert stated that parents are expecting change that will make computer science lessons more relevant to their children.

When explaining rationales for teaching CT, many articles in the literature mention the general benefits of CT as a thinking skill and the need to develop new skills for the employment market. At European Union level, the New Skills Agenda for Europe (European Commission, 2016) focuses on the need to develop digital skills for employability. The Agenda invites Member States to invest more in digital skill formation (including coding / computer science) across the whole spectrum of education and training. Employability and fostering the ability to live in our digital world are core rationales outside Europe as well. In the USA, President Obama's new initiative aims at equipping all students with "the computer thinking skills they need to be creators in the digital economy" and Computer Science is seen as a "new basic" skill necessary for economic opportunity and social mobility [80]. In Singapore, "a smart nation of the future, with a rich array of tech products and services, will require different skills". Hence, children will need to be more familiar with CT "as it becomes an increasingly

5 The Greek MOE did not reply to the survey questionnaire. However, the Greek Institute of Education Policy and the MOE Directorates for primary, secondary and VET education provided the CompuThink team with general information on the status of CT in the Greek education system.

essential part of our lives and careers". Finally, the new Australian curriculum[6] also emphasises the need for students to learn how to use and develop digital technologies in order to participate fully in the digital world.

**Table 4.** Rationale for integrating CT in the curriculum as emerged from the survey of MOEs

| | Austria | Czech Republic[7] | Denmark | Finland | France | Greece | Hungary | Italy | Lithuania | Poland | Portugal | Switzerland | Turkey |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fostering logical thinking skills | X | (X) | X | X | X | (X) | X | X | X | X | X | X | X |
| Fostering problem-solving skills | X | (X) | X | X | X | (X) | X | X | X | X | X | X | X |
| Fostering other key competences | | (X) | | X | | (X) | | X | X | X | X | X | X |
| Attracting more students into Computer Science | | | | X | X | | | | X | X | | | X |
| Fostering coding and programming skills | | | | X | X | | | | X | X | X | X | X |
| Fostering employability in the ICT sector | | | | X | X | | | | | | | | X |

In conclusion, in most countries - both within and outside Europe - the main rationale for introducing CT is to foster 21st century skills. These are seen as essential for active and fruitful participation in the knowledge society and, in a more pragmatic sense, for employment in a digitally-oriented jobs market.

The emphasis, however, varies. Some countries, such as Finland and Portugal, aim to reach quite specific goals, namely raising student achievements and increasing interest in mathematics.

## 4.2 CT in compulsory education curricula in Europe

The literature reports a recent upsurge of CT and, more broadly, of Computer Science in compulsory education. As part of curricula reforms, several Member States have already included – or are planning to include – CT and related concepts in compulsory schooling.

Although there is a great variety in the terminology used, as well as in the rationale and the strategies adopted, European countries[8] can be grouped into three clusters according to the approaches they are adopting to integrate CT and CS in compulsory education.

The first cluster includes those countries that, over the past three to five years, have started a process of curriculum review and overhaul. Irrespective of whether these processes were prompted by social challenges and job market needs, or were part of established reviewing procedures, they have all boosted the teaching of CT and related concepts in compulsory education at national level. In this strand, approaches to curriculum reform range from comprehensive and systemic overhaul to more focused adjustment strategies. The former approach generates an uninterrupted learning continuum that embeds the development of CT core concepts and skills in all students. These reforms are not limited to updating the curriculum but entail changes in teaching,

---

6 http://www.australiancurriculum.edu.au/technologies/digital-technologies/curriculum/f-10?layout=1
7 Countries which are still planning to integrate CT are in light blue.
8 As mentioned above, in the context of this study, we refer to Europe plus Israel and Turkey.

learning and assessment practices, and in school organization. Moreover, teacher training programmes and initiatives are also addressed to support the effective integration of CT and CS inside the curriculum. By contrast, the latter approach emphasizes and prioritizes specific CT concepts and skills, mostly related to fostering students' coding and programming abilities.

The second cluster encompasses those countries that have not yet begun introducing CT into compulsory education but are preparing to do so shortly. Among these countries, the level of policy discussion and decision-making varies from official studies advising governments to draft curriculum reforms.

The third cluster includes those countries in Europe that are building on their long standing tradition in Computer Science (CS) education, mainly in upper secondary schools. Among these countries, the main trend is to expand CS education towards lower secondary and primary level. In this perspective, CT plays a central role.

It should be noted that, for those countries where decisions are taken at regional level, it would not be accurate to provide a unified description of national policies. For these specific cases, where available, information was collected and included in the study only for the specific region concerned.

The distribution of the clusters in Europe and beyond, as emerged from collected evidences, is depicted in Figure 4.

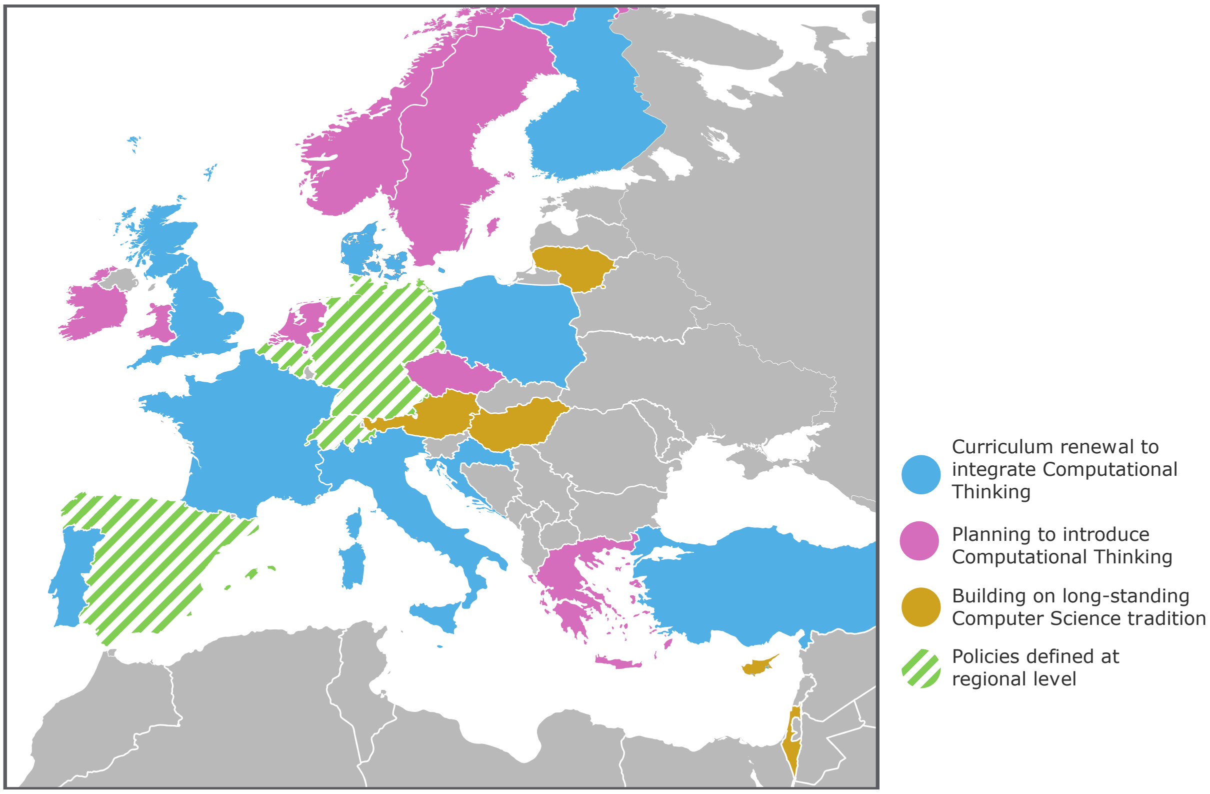



**Figure 4.** Prevailing approaches in integrating CT in compulsory education

### 4.2.1 Cluster 1: curriculum renewal process underway

A number of countries have recently concluded a process of curriculum renewal that has boosted the teaching of CT and related concepts in compulsory education at national level.

**England** (UK) has set an example here, being one of the first European countries to mandate CT and coding in primary and secondary schools (from September 2014 onwards). In the Computing programme of study [11], it is claimed that: "*A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world*." The decision of England's ministry to prioritise students' CT at all levels of school education paved the way to curricula reforms in different parts of Europe, with the consequent recognition of the relevance of CT and/or programming. England's pioneering effort is not just directed towards computation, but also to conceptualization of CT, which is strongly promoted in its educational agenda for compulsory education.

In 2015, the guidance and programming law no. 2013-595 of July 8th, 2013 for restructuring schools in **France** made provision for numerous reforms starting September 2016. In this context, the new common core, *Socle commun de connaissances et de compétences* [18], establishes proficiency benchmarks for all students completing the compulsory school cycle in France. The key concept of "algorithmic thinking" figures among the different languages foreseen in the guidelines, as students are expected to "*know about basic principles of algorithms and coding, they use simple programming languages*". Building on this document, the *Projet de programmes pour les cycles 2, 3 et 4* introduces digital literacy in primary and secondary schools, and foresees the teaching of algorithmic and programming concepts as a means "*to provide learners with a new language for thinking and communicating*" [19].

**Finland** is one of the first EU countries to introduce (as of autumn 2016) 'algorithmic thinking' (*algoritminen ajattelu*) and programming as a mandatory, cross-curricular activity from the first year of school (grade 1). The new National Core Curriculum for primary and lower secondary schools was published in 2014 and foresees implementation between August 2016 and August 2018 [15]. The new version of the Core Curriculum provides guidelines and learning objectives that relate to algorithmic thinking and programming foreseen as applicable in a transversal way. The innovative aspect of this approach is the creation of seven competence areas to be assessed as a part of subject assessment, thus combining competence-based and subject-based teaching and learning (see Section 4.3 Positioning CT in the curriculum). A significant aspect of the new core curriculum is the development of problem-solving skills in the context of real-life problems.

In turn, **Poland** counts on a long tradition of CS and informatics education, which has been part of school curricula for the last 30 years [41]. The curriculum followed until June 2016 features stand-alone informatics subjects, which were introduced to primary (grades 1-3) and middle schools in 2008, as well as to primary (grades 4-6) and high schools in 2012. Although the curriculum already included aspects of algorithmic thinking and informatics, a new unified Computer Science (Informatics) curriculum is to be tested from September 2016 and implemented in all schools providing compulsory education from September 2017. This new curriculum is in some parts an extension of the previous one, seeking to unifying aims at the different levels; applying a more homogenous terminology; and repositioning activities under the Computer Science umbrella [40]. The main goal of this new curriculum is to motivate students to apply CT (*myślenie komputacyjne*) and to engage in solving problems in various school subjects.

Other countries have also established policy initiatives that go in this direction. The *Piano Nazionale Scuola Digitale* in **Italy** [34] sets the government agenda for the nation to improve digital provision in education. The document explicitly mentions CT (*pensiero computazionale*), seen as a tool to move students from being passive users to active producers of technologies. A specific action is dedicated to programming as a way of bringing computational and logical thinking to all primary school learners, with the suggestion of adopting robotics activities for this purpose. According to the survey of MOEs, CT is part of the new national curriculum for primary and secondary schools in **Turkey** as well. Coding is also included (using block-based programming, text-based programming and robots).

In **Denmark**, CT is not a separate topic in K-9, but IT and Media is integrated across subjects in primary and lower secondary education. IT and Media includes skills such as problem-solving and logical thinking, but not all key CT characterization. It is planned to integrate Informatics as a compulsory subject, depending on school curricula, in grades 10-12 by 2017.

Similarly, in **Portugal**, CT (*pensamento computacional*) is mentioned among the learning outcomes (*metas curriculares*) for students at lower secondary level (grades 7 and 8). In 2015-2016, the Ministry of Education (ME) launched a pilot project for primary schools, entitled Introduction to *Programming in the 1st cycle of basic education* [43], involving 27,000 students in the 3rd and 4th years of schooling and about 670 teachers. The pilot focused on two main themes: Computational Thinking and Programming languages. The initiative has been extended to the 2016/2017 school year expecting to involve about 56.000 students and about 1.600 teachers.

In **Malta**, the government is setting out a vision to transform the country into one that will prosper as a digitally-enabled nation in all sectors of society[9]. This is also reflected in the *National Curriculum Framework of Malta* published in 2012 [37]. In this frame, the document entitled *Computing as a Core Entitlement Framework* [38], jointly developed by the Department of eLearning and the Department Curriculum – DQSE, outlines a practical strategy on how to introduce computing as a core entitlement for all students; this encompasses requirements for Digital Literacy, other literacies, as well as coding from early years. This document explicitly supports and guides the introduction of CT and problem-solving skills from kindergarten all the way to Year 11. In Malta, part of the learning outcomes framework will be implemented in 2017.

The curriculum reform started in February 2015 in **Croatia** as one measure of the *Strategy for education, science and technology* is expected to affect all levels of education, all subjects, cross-curricular topics and frameworks for assessment, special education and gifted education [4]. Within the new national curriculum, *Informatics* becomes an elective subject in all grades of compulsory education (previously addressed in grades 5-8) and an obligatory subject for two years in upper secondary education (*Gymnasium*). The new curriculum for Informatics builds on Croatia's long tradition in computer science and is organized in four domains, one of which is CT and programming (along with Information and digital technologies, Digital literacy and communication, and e-society). Experimental implementation of the new curriculum starts in the 2016/17 school year.

As emerged from the literature, in 2013, the government of **Scotland** enacted the new national *Curriculum for Excellence*, which introduced aspects of CT within the subject entitled "*Computing Science*" taught in secondary schools (secondary grades 3, 4, 5, 6) [46]. In September 2015, the Scottish Government began a four-month consultation to collect input for the development of a comprehensive approach to digital learning and teaching. The resulting document, a *Digital Learning and Teaching Strategy for Scotland*, was published in September 2016. Responses to the consultation from educational stakeholders call for the inclusion of CT at primary level [48].

### 4.2.2 Cluster 2: planning to introduce CT

Several countries in Europe are currently planning to introduce CT into compulsory education.

In the **Czech Republic**, the *Strategy of Digital Education until 2020* [8] sets out key priorities for initiating changes to the methods and forms of the Czech education system, as well as to its objectives. In this document, "*developing CT among students*" is one of the three priority objectives on which the first interventions will focus. Hence, CT is

---

[9] http://digitalmalta.org.mt/en/Documents/Digital%20Malta%202014%20-%202020.pdf

foreseen as a key digital competence all students need for their future lives, their professional careers, and for understanding the world around them. The curricular documents for both primary and secondary school education are due for release by the end of 2017; initial measures are to be piloted in selected schools from the 2018–2019 school year and are likely to be extended to all schools in 2020 [9].

The Ministry of Education and Skills in **Ireland** has defined a *Digital Strategy for Schools* [28], which provides a rationale and a Government action plan for integrating ICT into teaching, learning and assessment practices in schools over the next five years. The strategy calls for a wider definition of digital literacy for students, one which would include "*coding and programming in the Irish primary and post-primary curriculum so that every learner has an opportunity to learn skills such as CT, logic, critical thinking and strategic thinking to solve problems*" ([28], p. 22). The Irish Computer Society (ICS) developed two computing curriculum modules, Digital Media and Computational Thinking, which were piloted in 45 Irish schools between September 2012 and May 2013. Building on the pilot's success, the curriculum was expanded to four modules, which are freely available to schools [29].

In **Norway**, a special expert group evaluating the role of technology in primary and secondary education reported to the Norwegian Directorate in September 2016. The report [39] recommends a reform of the curriculum to include technology and programming (including CT) as compulsory subjects. There are plans on reforming the curriculum, but the outcome of the expert group's recommendation is uncertain, as the debate on whether to include programming and CT in compulsory education is still ongoing. Norway has started piloting the introduction of programming as an elective subject in 143 lower secondary schools, but no certain plans for compulsory education.

A new curriculum for **Wales** is currently under development, involving education professionals across the country [65, 66]. The new curriculum will have more emphasis on equipping students in compulsory education for life and it is expected to be enacted in schools by September 2018. The first element of the new curriculum to be developed is the Digital Competence Framework. This has four strands, one of which is named *Data and computational thinking* and includes the elements "*Problem solving and modelling*" and "*Data and information literacy*" [67]. From September 2016, schools are familiarizing themselves with the framework, formulating their strategic vision for cross-curricular digital competence and considering how to translate this into practice.

Similarly, **Greece** is also currently planning to include CT in primary and secondary education curricula. A recent report, prepared by the Committee of Continuous Educational Affairs of the Greek Parliament and published in May 2016, suggests including CT in the curriculum as a short-term priority [23]. It also suggests implementation from the first year of primary to final year of secondary school, although the actual scope of the implementation has not yet been decided.

In the **Netherlands**, a wide-scale debate is presently underway about including CT in the *Information literacy* subject, which is already part of the core curricula [63, 64]. In 2012, The Royal Netherlands Academy of Arts and Sciences (KNAW) published a report on *Digital Literacy in Secondary Education* containing a number of recommendations on digital literacy and CS. One of these suggestions is for CT to play a central role in a new digital literacy course and in a revised CS course. As Joke Voogt pointed out during the expert interviews for this study, the Netherlands currently has an optional subject in upper-secondary school called Informatics but this is not offered in lower-secondary and primary education. Dutch schools exercise a fair degree of autonomy and some have acted independently in this direction, particularly by introducing programming, including at primary level.

Likewise, in **Sweden** the Government has recently requested the Swedish National Agency for Education to come up with suggestions for changing the national curricula, including the strengthening of the digital competence and programming. A proposal is expected by summer 2016 [60].

### 4.2.3 Cluster 3: building on a long-standing CS tradition

Several countries are moving forward by building on a long-standing tradition in Computer Science.

In **Austria**, the development of curricula for secondary schools in the area of Informatics includes CT-related concepts such as modelling and abstraction and takes problem solving as its central aim. Students are expected to develop understanding of the theoretical foundations and get to know basic principles of machines, algorithms and programs [1, 2].

Computer programming and coding are part of the computer science curriculum in **Cyprus** [5, 6, 7]. As of 2001-2003, lower secondary students have been introduced to algorithmic thinking and programming, which is compulsory for students aged 13-16. There is no distinct computer science subject in the primary school curriculum, but computers support other learning.

**Israel** has a long tradition in Computer Science education. Although CS is offered as an elective subject in most of high schools, students in mainstream public education are instructed in digital and computer literacy as a major medium and methodology contributing to learning in all subjects [31]. The Ministry of Education in Israel has a well-organised and detailed strategy for the study of Computer Science in high schools. This regards CS both as a stand-alone subject and the promotion of digital and computer literacy as a priority across all subject areas [30, 31]. The curriculum consists of both mandatory and elective modules. The Introduction to CS, for instance, emphasizes the fundamentals of algorithmic thinking. The intention of the CS course in not to train students to become programmers but rather to introduce learners to logical and algorithmic thinking and to expose them to different development environments at an early stage [32]. By offering a range of mandatory and elective modules, the curriculum allows all students to acquire the foundations of CS, while providing more time and content for those who have a more specific interest in CS [33]. In recent years, a new program was introduced for middle schools (grades 7-9) and a program to teach CS in primary school (grades 4-6) started this year school year (2016-2017).

In **Lithuania** CT is integrated in the Information Technology subject, which is compulsory in lower secondary schools (grade 5-10). IT includes five knowledge areas: information; digital technologies; algorithms and programming; virtual communication; security, ethics and legal principles. At the upper secondary level (grade 11-12), IT is an elective subject offered in basic and advanced modes. The advanced course includes electronic publishing, database design and management, and programming [36].

The National Core Curriculum of **Hungary** (2012) includes algorithmic thinking as a competence for primary and secondary education inside Information Technology. Informatics is a compulsory subject from grades 6 to 12, with the goal of teaching logical and algorithmic thinking, and problem solving [24, 25, 26]. In October 2016, the Government adopted the Digital Education Strategy [27], which includes goals regarding the "digitalization" of all educational sectors, including primary, secondary, higher, vocational and adult education, mainly to be reached by the end of 2018. The current draft suggests concrete targets regarding the integration of Computational Thinking/programming into school education, but final decisions are yet to be taken. For instance, the Strategy indicates the ability to teach programming among the requirements for Math and Science initial teacher education programmes. Furthermore, it also suggests to revise the curriculum of the Informatics subject including coding/programming starting from 3rd grade on, as an individual subject.

In **Slovakia**, Informatics is currently a mandatory subject at all level of compulsory education. It was introduced at upper secondary level in 1985, at lower secondary in

2005, and at primary in 2008. Programming has always been one of the key components of this subject 49, 50].

### 4.2.4 Policy initiatives at regional levels

In some countries, curricula are developed at regional level, therefore the integration of CT varies from region to region.

For instance, information from the survey of MOE indicates that in **Spain** no specific official national documents mention "computational thinking". Nonetheless, some connections in terms of concepts can be found in the curriculum of specific subjects. At a national level, in compulsory secondary education the subject "Technology" is included in the curriculum. Similarly, the academic upper secondary studies include the subjects *Industrial Technology* and *Information and Communication Technologies*, whose curriculum is further developed in each autonomous community. Some similar connections can also be seen in subjects only offered in specific autonomous communities, including the Autonomous Communities of **Andalucia**, **Canaria**, **Cantabria**, **Castilla – La Mancha**, **Castilla y León**, **Región de Murcia**, **Madrid**, **La Rioja** and **Comunidad Valenciana** [51, 52, 53, 54, 56, 57, 58, 59]. As emerged also from the desk research, in the Autonomous Community of **Catalonia**, aspects of programming are included in the subject digital competence, which is compulsory for primary schools. In secondary schools, aspects of programming and robotics are present in the teaching of ICT.

According to the literature [22], in 2004 the state of **Bavaria** in **Germany** introduced a new compulsory subject of computer science (CS) in its lower secondary schools (Gymnasium). The subject is based on a comprehensive teaching concept that builds upon a long tradition in CS teaching. It comprises mandatory courses in grades 6 & 7 for all grammar school students and in grades 9 & 10 for students who attend schools offering a science and technology track. At upper secondary level (grades 11 and 12) there are elective courses that qualify students for an optional graduation exam in CS. In the **North-Rhine Westphalia** region [21], lower secondary schools offer courses in ICT and basic CS concepts. Digital literacy is taught through other subjects, usually in grades 7 or 8. A common theme among these curricula is to relate CS topics to relevant contexts and practices outside the classroom. Currently, CS education in lower secondary schools focuses on ICT skills (*Grund-bildung Informatik*).

In 2007 the **Flemish** government in **Belgium** issued a set of ICT educational standards which should be achieved at the age of 14 (the end of grade 8). Schools are advised to integrate these standards in their teaching across the curriculum in grades 5-6 and 7-8. Part of the Informatics course in grades 9 and 10 currently focuses on enhancing digital literacy. Since last year, primary school children should be attending an integrated course of science and technology, which contains elements of CT. Students in grades 11-12 currently have available two study profiles with significant CS content: IT & Networking, and Accountancy & IT. Both profiles, however, are part of professional studies in Economics and Administration [3].

In the **German speaking part** of **Switzerland**, the curriculum for primary and lower secondary schools lists competences which are part of - or lead to - CT, including coding and programming. CT is addressed within computer science as well as in other subjects. At upper secondary level, a national curriculum framework for non-vocational schools is in place. CT and coding/programming are mentioned among the objectives. In the **French speaking part** of Switzerland, CT comes under a specific part of the *Plan d'études romand* (PER) called MITIC (Média, image, technologie de l'information et de la communication). MITIC aims at developing general digital learning [61, 62].

## 4.3 Positioning CT in the curriculum

In the following, we examine the position of CT in curricula, considered in terms of two criteria: *level of education* and *subject*. Looking at CT integration in compulsory education, we can see that most countries integrate CT at secondary level. However, a growing trend towards integration in primary schools is also emerging (Figure 5).

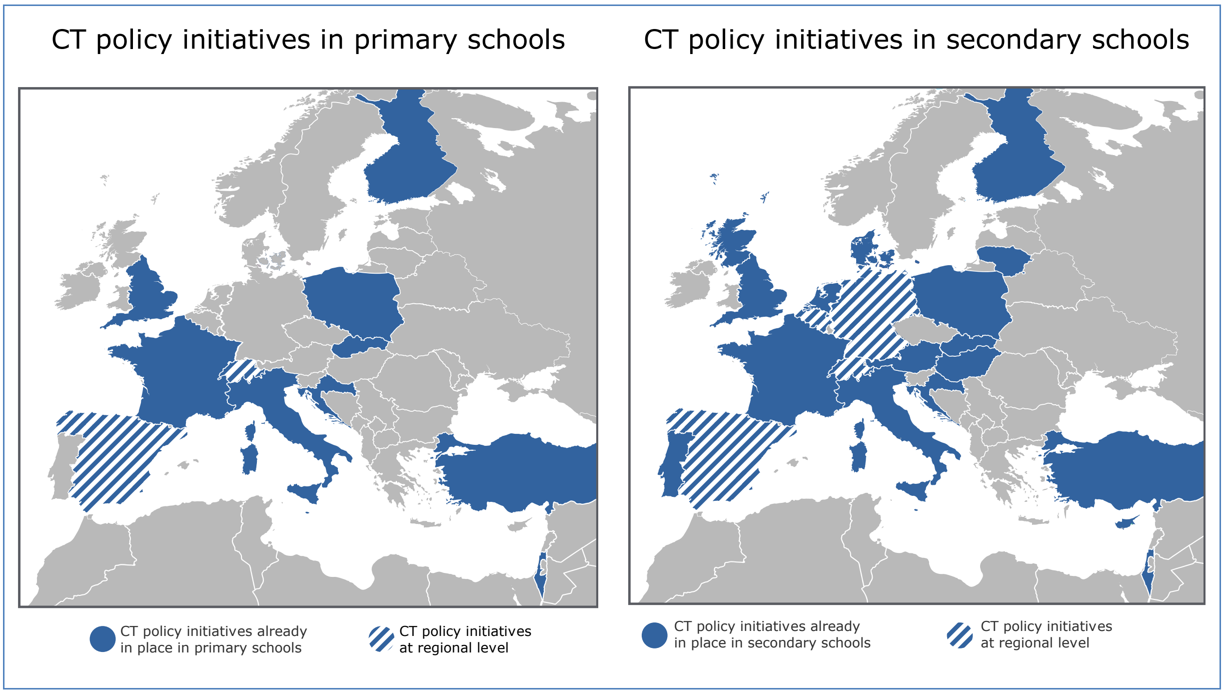



**Figure 5.** CT integration by education level in Europe and beyond (Turkey and Israel)

An important issue in the integration of CT in compulsory education is its position in the curriculum: should it be a subject in its own right or embedded across other subject areas?

In **Finland**, the formal teaching of programming and algorithmic thinking is part of maths (grades 1-9) and crafts (grades 7-9). In grades 1 and 2, students learn about the principle of giving step-by-step commands. Subsequently, in grade 3, they start using visual programming tools. In the final years of basic education (grades 7-9), they gradually progress from simple to more complex tasks, learning what algorithms are and comparing the usefulness of different algorithms. However, programming is applied to all subjects as a means and as a practical activity. It also supports several of the seven transversal competences in the national core curriculum, especially: Thinking and learning to learn (transversal competence 1); Cultural competence, interaction and expression (transversal competence 2); Multi-literacy (transversal competence 4); ICT-competence (transversal competence 5); and Competence for the world of work, entrepreneurship (transversal competence 6).

In **France**, the understanding and creation of algorithms underpins active engagement with technologies and programming as part of maths studies. During the first school year (cycle 2), as part of developing their understanding of the world around them, students learn to code movements in space using suitable software; this leads in the second year to the understanding and production of simple algorithms. Cycle 3 focuses on progression towards abstraction in all domains. In this cycle, pupils are formally introduced to programming. In Cycle 4 (lower secondary school), algorithmic thinking is the basis of the development of logical thinking, and informatics teaching is split between maths and technology.

In **Portugal**, CT is part of ICT and Informatics subjects. Algorithm and programming concepts are taught to 7th and 8th graders as part of a compulsory subject called ICT. These topics are also addressed in Informatics Professional Courses (10th to 12th grades – upper vocational secondary school) as elective courses and an optional subject for the scientific courses.

In **Austria**, CT and related concepts are part of the subject Informatik taught in upper secondary school (general and vocational education). Students are expected to learn theoretical fundamentals and get to know basic operational principles of machines, algorithms and programs [1].

Table 5 summarises how CT is located in the curriculum based on the results from the survey of MOEs.

**Table 5.** Curriculum location based on the survey of MOEs

| Country | Within a subject | Across all subjects | Depends on regional or school curricula |
|---|---|---|---|
| **Austria** | Informatics (upper secondary level) | | |
| **Denmark** | Information/technology (in grades 10-12) | (in grades 0-9) | X |
| **Finland** | Mathematics (grades 1-9)<br>Crafts (grades 7-9) | Transversal competences (e.g. ICT competences) | X |
| **France** | Mathematics (Cycle 2-3, primary level)<br>Math and Technology (Cycle 4- lower secondary) | | |
| **Hungary** | Information technology (grades 1-4; and grades 9-12) | | X |
| **Italy** | Informatics/ technology<br>IT Curriculum - Applied Science | X | X |
| **Israel** | Computer Science | | X |
| **Lithuania** | Informatics and Information Technology (IT) (grades 5 -12) | | |
| **Malta** | ICT subject | Part of Digital Literacy (primary level) | X |
| **Poland** | Informatics (grades 0-12) | X | |
| **Portugal** | - ICT subject (grades 7-8)<br>- Informatics (grades 10-12) | | |
| **Switzerland** | X | X (primary and lower secondary level German speaking schools) | |
| **Turkey** | ICT and Informatics (grade 5-6) | | |

## 4.4 Examples of CT integration in compulsory education around the world

In July 2016, the Education Minister of **New Zealand** announced that digital technologies will be fully integrated into The *New Zealand Curriculum and Te Marautanga o Aotearoa* (national curriculum for Māori-medium) from 2018 [72]. Digital technologies will be included as a part of the existing Technology area of the national curriculum for grades 1-13. They will follow six themes: algorithms; data representation; digital applications; digital devices and infrastructure; humans and computers; programming.

Computing is a specialized learning area in the new **Australian** Curriculum launched in 2015. In this curriculum, *Digital Technologies* is addressed as a whole discipline (compulsory in K-10), where the actions and interactions of humans and computers is of as much importance as the specific knowledge and skills required to think computationally [69]. Another subject, *Design and Technologies*, complements instruction on the topic. Both subjects provide opportunities for students to create solutions, develop a range of thinking skills (including systems thinking, design thinking and CT), learn how to manage projects, and consider how solutions that are created now will be used in the future [70]. The curriculum is mainly centred on problem solving and algorithms.

The **South Korea** *Software Education* program, currently in its pilot phase, is focused on developing CT, coding skills, and creative expression through software. It is due to be rolled out at all levels of education: primary, secondary and university. Primary and lower secondary will face the most dramatic change because the new programme will be mandatory at these levels beginning in 2018. Training for primary teachers is especially critical to the success of this policy since elementary school teachers teach all subjects and there are no separate IT/computer teachers. By 2018, 60,000 elementary school teachers (30% of the total) will receive specialized training in software education, and 6,000 of that trainee population will receive in-depth training. In addition, 1,800 middle school teachers who are certified to teach IT/computing will receive additional training on software education [77].

The 2016/17 school year in **British Columbia, Canada** begins with the launch of the officially redesigned curriculum for primary and lower secondary education (K–9). This is part of a three-year transitionary process, which commenced in the autumn of 2015 [71]. CT is integrated as a Core Content from grades 6 to 8 as part of the subject *Applied Design, Skills and Technologies*. The focus is on simple algorithms that reflect CT, visual representations of problems and data, the evolution of programming languages, and visual programming. A draft curriculum for upper secondary level (grades 10–12) is available and ready for optional use in classrooms in the 2016/17 school year.

Singapore and Japan are also moving in this direction. **Singapore**'s aim to be a *Smart Nation* has prompted 19 secondary schools to offer programming as part of a new Ordinary Level subject called *Computing*. This will start in 2017 at the Secondary 3 level and will replace the existing *Computer Studies subject* offered by 12 secondary schools. The move puts into action a call made by Singapore's Prime Minister in 2015, when he launched the Smart Nation initiative, for children to be exposed to programming from a young age. The new subject will focus on programming, algorithms, data management and computer architecture [76].

Similarly, the Education, Culture, Sports, Science and Technology Ministry in **Japan** has recently announced it will make computer programming a compulsory subject at primary schools as of 2020, followed by middle schools in 2021 and high schools in 2022. Programming is defined as creating software programs that work in the way intended by programmers [78].

# 5. Approaches to CT teaching, learning and assessment

CT is being advocated as a key 21st century skill which should enable students to be not only technology-literate but also creators of computational artefacts. Making room for CT concepts and related skills in compulsory schooling ought to enable students to use computational tools to express themselves, solve problems, represent knowledge and construct models and simulations.

## 5.1 Pedagogical approaches

The experts who were interviewed agreed that multiple pathways to CT should be used throughout compulsory education. In particular, Mitchel Resnick pointed out that it is crucial to give learners the opportunity to design, create and experiment in areas they care about. Joke Voogt concurs on the need to identify effective pedagogical approaches for fostering CT in compulsory education that draw on real-life situations. Other elements that emerged from the interviews are context awareness and teacher readiness. Furthermore, a number of interviewees questioned the current uptake of coding: Judith Gal-Ezer, for instance, points to the fact that over-reliance on coding might give pupils a false impression of what CT is. Simon Peyton Jones discussed the need to undertake research to evaluate the effectiveness of different approaches.

An extremely popular approach, commonly adopted in many countries, is the CS Unplugged, whereby computing is taught without using technology (e.g. Curzon et al., 2014). Unplugged activities involve problem solving to achieve a goal and, in the process, deal with fundamental concepts from CS. The integration of physical activity in this process makes it lively and engaging. A typical example is the sorting network (Bell et al., 2012), where a layout like the one in Figure 6 is drawn on the pavement. Students hold numbers and are positioned in the squares on the left. They then move in the direction of the arrows. Students meet two by two at the circles (nodes) and compare the numbers they hold. The student with the smaller one follows the arrow to their left, while the student with the larger number follows the arrow to their right. They then meet other students at the subsequent nodes and compare numbers again, continuing the process until they reach the squares on the right. "*Students and teachers alike are generally surprised when they come out the end of the network with the numbers they are holding in ascending order*" (p. 401).

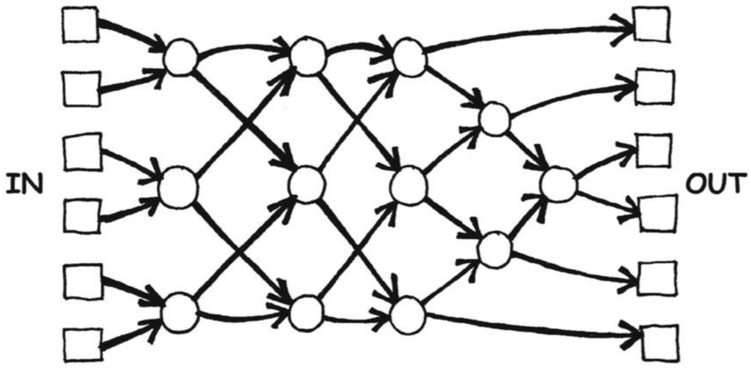



**Figure 6.** A six-input sorting network layout (from Bell et al., 2012, p. 402)

Computer simulations are often used in science classes to support learning. Learners use simulations to explore phenomena, engaging in "what if" experiments and reflections while changing the values of the simulation's parameters. Some proponents advocate student development of simulations as a strategy in science education but the complexity of the maths that this entails (i.e. algebra and calculus) represents a serious obstacle. Computational models, unlike the corresponding maths representations, are executable models that can be more easily tested, debugged and refined. Familiarity with CT & programming skills might enable students not only to use simulations, but also to modify the underlying computational model and eventually design and implement their own model and get it to run a simulation. Project GUTS is an example of implementing a "use, modify, create" learning progression in computational modelling and simulation at secondary school level (Lee et al., 2011).

Research in the use of computational modelling in science education provides evidence that this approach is more learnable (Weintrop et al., 2015). Scalable Game Design (Repenning et al., 2015) advocates starting from a computer game construction project in order to reach computational modelling and simulation in STEM. Scalable Game Design builds on the motivational aspects of game design to foster a transfer of skills from game design and implementation to simulation and modelling via Computational Thinking Patterns (CTP). CTP are design patterns acquired in constructing computer games and later transferred to the creation of STEM simulations (Ioannidou et al., 2011). This resonates with the view Joke Voogt voiced in the interviews that transfer does not happen *per se*, rather it needs to be addressed explicitly in teaching[10]. This is borne out in formal research and practical experience, including work with Logo (Papert, 1980), which clearly indicates that transfer only happens if it is part of the pedagogy.

Several authors also highlight that, when introducing CT in compulsory education[11], there is a need to adopt an inclusive approach addressing gender equity and special education needs. A study by Atmatzidou and Demetriadis (2016) on educational robotics with 15 and 18 year old students concluded that when the overall instructional context is supportive and learning activity time is adequate, all students may overcome their initial difficulties and successfully develop their CT skills. A study by Snodgrass and colleagues (2016) suggests that "*primary teachers could use their professional judgment related to how to support students with disabilities in other content areas (e.g. reading, mathematics) within the context of computing instruction. Thus, finding effective supports required knowing the individual support needs of students and, when applied in CT, were feasible and effective even though the teachers were still developing their understanding of CT pedagogy*" (p. 16).

[10] See Box 2 for a few examples of current research on CT and transfer of knowledge.
[11] See Box 3 for the view of experts, interviewed in this study, on an inclusive approach to CT.

**Box 2.** Transfer of knowledge

The claim that the computer culture might shape children's thinking, thus becoming an object to think with, has an historical precedent in the work of Seymour Papert. Papert held that, by programing the computer, children would get in touch with powerful ideas and thereby enhance their learning. This bold position has stimulated a trend of research seeking evidence for the transfer of cognitive skills from programming activities to other domains, for example mathematics. The transfer of cognitive skills (e.g. problem-solving abilities) is key to the claims for introducing CT in compulsory curricula.

Gaining a more comprehensive picture of CT skills requires additional assessments, such as a test of students' knowledge transfer, or the collection of more in-depth, qualitative data from both students and teachers. Grover, Pea and Cooper (2015) focus on multiple assessment mechanisms or a "system of assessments" to provide a more comprehensive view of student learning than commonly used CT assessments, such as evaluation of student-created programs. The *Computational Thinking Pattern Analysis* (CTPA) framework (see Section 5.1) has produced early indicators of transfer from game design to computational science. The key assumption of the CPTA framework is that if students can build games using Computational Thinking Patterns, it might be possible that they can apply these same patterns in the implementation of science simulations. By automatically breaking down complex programs into constituent parts, CPTA is providing ways of measuring these patterns in the artefacts students build. CPTA is beginning to show evidence of the transfer of skills between domains.

**Box 3.** An inclusive approach to CT

The declared goal of the EU digital agenda of creating an information based society calls for equity and inclusiveness when integrating CT in compulsory education. This is particularly important in the field of Computer Science and related careers, where the rate of underrepresentation connected to gender inequality (Sax, 2012) and race gap (Margolis, 2008) is higher than in STEM.

Approaches like Exploring Computer Science (ECS)[12] look particularly promising for involving underrepresented minorities in upper secondary education. The ECS initiative includes both the curriculum and a teacher professional development program. The ECS curriculum was designed to engage all students in computational thinking, with a special focus on underrepresented students in low-resourced schools. The ECS teacher professional development program focuses on inquiry and equity-based practices (Ryoo et al., 2016).

Similarly, interviews with experts also highlighted that an inclusive approach should be adopted when implementing CT in compulsory education. Voogt points out that CS careers are mostly pursued by boys, and that CT should be implemented in an attractive way and with creativity. Yongpradit reports that in the US there is a gender issue in relation to learning technology, which is found interesting mostly by boys, in even greater measure than science. According to Resnick, programming in itself is not biased one way or the other. However, the way it is introduced can be biased. For example, a tutorial on how to make a videogame will probably engage boys more than girls. Learning how to create interactive imagination stories is likely to attract more girls. Hence, different kinds of activities appeal to different types of kids, making it imperative to provide multiple pathways to CT.

Dealing with students with special needs and learning disabilities, Ruth Sanders highlights the opportunities offered by both physical and virtual tools (e.g. BeeBot and ScratchJr) to create an inclusive teaching and learning environment, where all pupils in the class are engaged in programming activities. According to Ruth Sanders:

*"All pupils, regardless of their ability, will benefit from coding, largely because they can work at their own pace and level according to their individual needs. Coding is engaging for students because they get involved, they are keen to experiment and try things out. There is no obvious right or wrong method. If their code does not work, pupils just continue experimenting until they achieve what they want to achieve. They can either create their own game or they can just do basics like moving a character on the screen."*

[12] http://www.exploringcs.org/

## 5.2 Learning tools

The introduction of CT core concepts and skills in compulsory education requires learning tools that can make programming activities accessible to young children in primary school (i.e. tools that have a low floor) and yet be challenging for more experienced learners (i.e. high ceiling).

Programming can make CT concepts concrete, opening the way for learning of powerful ideas. In his interview for this study, Simon Peyton Jones points out that learners should not only be able to write programs but also to read them. However, learning to read and write a new language is no easy matter.

The pioneering work in making programming accessible to children was done by Seymour Papert, who introduced the Logo programming environment for novices back in the sixties. This had some key design features that are still present in modern programming languages for children. His "turtle robot" is both physical and virtual (screen-based). The Logo commands for controlling it (e.g. forward, right) are body syntonic, i.e. the child can step into the turtle's shoes to execute them. Thus its movements (on the floor or the screen) provide visual feedback. Logo is text based; the first drag & drop block-based interface was introduced in 1996 with LogoBlocks, the programming environment for an early prototype of Lego Mindstorms. More recent block-based environments like Alice (Figure 7) and Scratch (Figure 8) feature multiple screen avatars. Both tools allow novices to focus on creating and experimenting, as they do not require the learners to be able to code in a textual language (e.g. Duncan et al., 2014).

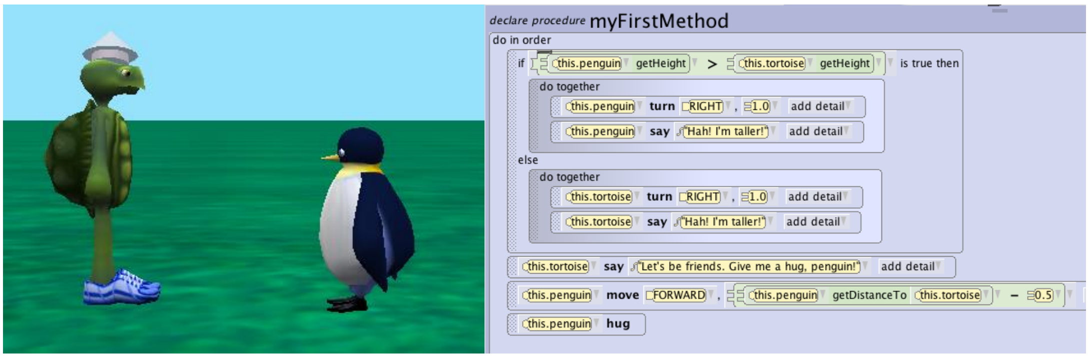



**Figure 7.** Code in Alice controlling interaction between a tortoise and a penguin


**Figure 8.** Scratch code for a poking game

A popular application domain for visual programming is learners' creation of digital games. While drag & drop visual languages can be easy to start with, they can also be quite complex and sophisticated. For example, to control multiple avatars and interactive animations, the programming language includes primitives for concurrency and handling of events. Figure 8 shows a videogame where the user has ten seconds to click on the cat as it moves to different random positions on the screen. The game is implemented with three scripts that execute concurrently.

Block-based programming environments are also used to carry out activities of animated storytelling, in which the user has to decompose scenes and characters' movements in a similar way to what is done in game creation. All these kinds of activities - game creation, robot programming and storytelling - have been proposed for different educational levels, obviously with different complexity. Several other learning tools for introducing CT in compulsory education emerged from the desktop research. In addition to block-based environments (e.g. Kodu, Greenfoot, Agentsheets, Agentcubes), these include tangible tools like robotics kits (e.g. Lego Mindstorms), e-textiles (Lilypad) and handheld computers (e.g. BBC micro:bit).

In young children's education, the programming of toy robots (e.g. Bee-Bot) is also widely applied (e.g. Atmatzidou & Demetriadis, 2016). For this activity, in particular, the learner needs to split the actions (that wants the robot is to carry out) in a sequence of movement, paying attention to spot similar actions in different situations that can be repeated without re-programming them. Hence, the learner carries out useful practices of abstraction and decomposition. This resonates with the view Resnick voiced in the interviews about different affordances of physical and virtual environments supporting multiple pathways to CT: "*With the physical environment it's easier to have many kids in the same space working together. With the virtual it's easier to share things and collaborate at a distance. Some kids will have a natural attraction to make things in the physical world. Others will take advantage of the affordances of the virtual.*"

## 5.3 Assessment

Several authors represented in the desk research recognize that assessment of CT concepts and practices is essential for full and effective integration of CT in education (e.g. Grover et al., 2014). However, the range of research works dealing with assessment of CT concepts and constructs - and their transfer to other knowledge domains - remains quite limited.

Brennan and Resnick (2012) describe three main approaches to assessing the development of CT:

- analysing students' portfolios of projects and generating a visual representation of the (programming) blocks used (or not used) in each project;
- artefact-based interviews, based on discussion of two interviewee-selected projects;
- design scenarios - given a set of three projects with low-medium-high complexity levels, the interviewee is asked to select one and (1) explain what the selected project does, (2) describe how it could be extended, (3) fix a bug, and (4) remix the project by adding a feature.

Most strategies assess CT by analysing the artefacts (e.g. games or models) that students develop as indications of their CT abilities. One strategy for measuring CT requires students to modify the code of an existing program so as to accomplish specific objectives. Troubleshooting scenarios, i.e. debugging an existing program, could also be an effective way for assessing students' fluency in computer programming and computer-based problem-solving.

Several authors also report the use of multiple-choice assessments and attendant rubrics to assess CT skills of middle school students. A recent CSTA study (Yadav et al., 2015) summarizes what is known about assessment of student learning in high school Computer Science (CS) in the USA. Similarly, Simon Peyton Jones reports in the expert interviews that the Computing At School initiative has started a project called Quantum on assessing computing in primary and secondary schools. Project Quantum will provide free access to an online assessment system that helps computing teachers to check their students' understanding and support their progress.

Tools have been developed that support educators' evaluation of student programming and assess the development of CT. One example is Dr. Scratch, a tool that performs automatic analysis of Scratch programs, detecting the presence/absence of specific primitives (e.g. conditional statements) in students' work. As well as providing feedback to educators and learners, Dr. Scratch assigns a CT score to analysed projects (Moreno et al., 2015).

Alongside more traditional multiple-choice tests and open-ended questions, a design-based approach (i.e. programming interactive media) emerges as an essential element of assessments systems.

Assessment is a crucial aspect of CT in education that is still underdeveloped. The interviewed experts concurred that assessment of CT is at an early stage, with some pointing to the need for further research (Voogt, Lepeltak). Gal-Ezer, Peyton Jones, Lepeltak, Urschitz and Voogt converge in recognising that current assessment methods and tools only cover some discrete aspects of CT. Peyton Jones affirms that we are currently unable to assess the whole spectrum of CT across age groups.

# 6. Training teachers in Computational Thinking

The introduction of CT in compulsory education requires support measures to prepare teachers. According to Eurostat, there is a total of 2 million primary teachers and 2.5 million secondary teachers in the 28 EU countries. The introduction of CT into the curriculum at all educational levels is creating demand for large-scale in-service continuous professional development. For example, in Italy the MOE *Digital School Plan* foresees training for 157,000 teachers, from March 2016 to December 2017, through blended training, workshops, online training, and cascade training. In France, the *Class'Code*[13] teacher training project led by the SIF (Société Informatique de France) and managed by INRIA (the French national institute for computer science and applied mathematics), estimates that 300,000 teachers will be involved in professional development on CT. *Computing At School* (CAS) proposes an innovative approach to support the new computing curriculum in England: experienced teachers become *CAS Master Teachers*, who work with around 40 teachers in their local community. These CAS Master Teachers receive a 5-10 days training course over six months. There are currently over 350 Master Teachers, with the goal of recruiting 600 by 2018.

Similar efforts are being deployed and/or planned worldwide. The US National Science Foundation financed the *CSK10 program*, which from 2010 to 2016 trained 10,000 upper secondary CS teachers. To implement its new Software Education programme (see Section 4.4), South Korea's Ministry of Education and Ministry of Science, ICT and Future Planning will provide specialized software education training to 60,000 primary school teachers, with 6,000 receiving in-depth training.

Grassroots efforts are also contributing to teachers' professional development. For example, *Code.org* trained about 30,000 teachers in US compulsory education over the last three years, organizing professional workshops and holding conferences for teachers and teacher trainers.

The teacher training opportunities discussed in the literature largely focus on pedagogical aspects rather than technological skills. Most training seems to be designed for all subject teachers, sometimes with a particular focus on STEM teachers. Pedagogical approaches addressed include storytelling, problem solving, deductive and inductive pedagogies with a focus on computational models and simulation. Often, training activities are designed specifically to be hands-on so that teachers can more easily transfer their new skills to their classrooms. While several MOOCs have been developed, a face-to-face component of teacher training is still relevant. A recent survey of over 900 in-service teachers in England concluded that face-to-face events and training, paired with an online community, are considered to be particularly effective in addressing their needs in content knowledge and pedagogical content knowledge related to CT (Sentance & Humphrey, 2015).

While only a few articles in the desk research specifically focus on training for pre-service teachers, four interesting approaches are worthwhile highlighting. The first approach is

13 https://project.inria.fr/classcode/

the *Partner4CS Professional Development* model that includes not only a summer institute, but also follow-up classroom-based support and online support (Mouza et al., 2016). The second one is the integration of CT in existing modules on problem solving and critical thinking within a required educational psychology course for pre-service teachers (Yadav et al., 2014). A third approach comprises a series of pre-professional development interventions to assist teachers in utilizing CT and programming as an instructional tool within other subject areas (i.e. music, language arts, mathematics, and science). In the fourth approach, trainee teachers use the Flash Action Script to write pseudocode to solve a problem (CT skill) and translate that pseudocode into Action Script (programming skills). The training shows how the Action Scripting language could be used to produce a teaching artefact; at each stage the participants wrong thinking was identified and corrected (Saari et al., 2015).

All the interviewed experts discuss or at least mention teacher training. Voogt suggests adopting a multi-perspective approach in preparing teachers, as we are faced with multiple issues: first, the specialisation of teachers for upper secondary level; second, the array of competences of teachers at primary school level. Voogt explains that many countries have CS teachers at upper secondary level, but too few at lower secondary and primary levels. Gal-Ezer reports that, at times, Israel faced a shortage of specialists even to teach in high school and it was necessary to train teachers of other subjects to teach CS. Most of these teachers went through a crash program, taking about ten courses that form the basics of computer science. Lepeltak calls for a professionalization of teachers who are asked to impart CS lessons. At primary level, Voogt argues, there is a need to make room in teacher education programmes for computer science specialists who can teach at least basic notions of CT, probably related to STEM subjects.

**Box 4.** Professional development of teachers in CT in England

In England there are more than 500,000 teachers in compulsory education. In secondary schools the new "computing" subject is assigned to the 14,000 existing ICT teachers, who have to be up-skilled. In primary school there are around 200,000 teachers, who teach all subjects to a particular class and are to take on computing.

The teacher training task in England is being addressed by Computing At School (CAS), a grassroots initiative with financial support from the Department for Education. CAS has established the Network of Teaching Excellence in Computer Science (NoE) for supporting, training, and equipping teachers as they implement the Computing programmes of study in their classroom. Simon Peyton Jones describes the NoE approach as follows:

> "*The NoE has selected and trained Master Teachers, i.e. experienced classroom teachers with a passion for the subject; enthusiasm, energy, and a desire to support others. Master Teachers, with the support of their Head Teacher, are expected to dedicate one afternoon a week to train other teachers in their area. We develop some learning material centrally (e.g. QuickStart Computing), but is up to the master teacher how to run the training. Now, we have 350 Master Teachers active in the NOE. Starting from September 2015 we have introduced 10 Regional Centres based in Universities, involving either the Computer Science department or the School of Education and in some cases both, working with the CAS Master Teachers in their area to promote and support relevant teacher engagement and CPD activities*."

Overall, in England there are approximately 24,000 schools, including 16,800 primary schools, 3,400 secondary schools and 2,400 independent schools (primary and secondary). The NoE's plan is to recruit 600 Master Teachers by 2018 each supporting 40 local schools by designing and running not-for profit CPD activities for those schools.

At the same time, Gal-Ezer warns that teaching CS to young children requires specific pedagogical approaches and teaching capabilities. Even though it is an advantage to have specialized CS teachers at primary and lower secondary levels, the main concern here is that those teachers should be capable of approaching CS in a fashion that is suitable for young pupils.

Voogt and Lepeltak concur in considering that teacher training could also be pushed at EU level, by joining forces on the professionalization and training of teachers. Voogt, Pahkin and Grečnerová mention MOOCs as possible means for teacher professional development, although Voogt warns about controversial aspects of online courses. Lee and Resnick point out the importance of offering teachers the same experience as students. This approach allows teachers to follow the same path as students, so that they can get an idea of implications and potentials. Voogt refers to the lack of resources or expertise in many countries for teacher training.

Finally, Vitikka recommends to involve teachers in the curriculum renewal process as a mean of professional development. This, in fact, would allow teachers to get gradually engaged in the reform, familiarize with the main ideas and don't perceive the curriculum change as a top-down process with imposed guidelines and regulations.

# 7. Beyond formal education

There are clear signs that MOEs in Europe and beyond are intensifying efforts to integrate CT, programming, computing, algorithmic thinking, CS and coding in formal education, albeit using a variety of approaches. At the same time, a notable array of initiatives is emerging in non-formal or informal settings at local, national and international levels. Hereafter we provide an overview of the initiatives we have encountered. They are grouped according to their current reach and location: global level, for those initiatives that are present on more than one continent; European level, for initiatives present either at Pan-European or member State level; extra-European level, for initiatives outside Europe. We should underline the fact that this list is not exhaustive, as a comprehensive global survey was beyond the scope of this study. We are certain that many more initiatives currently exist inside and outside Europe from which valuable lessons can be learnt.

Initiatives arising outside formal education have been the first to fill the perceived gap between social needs for computing and CT skills and educational provisions, and some have quickly reached global status. As initiatives generally developed in non-formal or informal settings, they are not necessarily tied to curricular constraints, and tend to foster a participatory technological culture.

As highlighted by Mitchel Resnick during the interview: "*In general, formal settings can provide structure for systematic thinking and approaches, while informal settings can help children build up motivation and identify their interests. An ideal setting should have both of those*." Although many initiatives were originally more coding-oriented, almost all are currently taking a turn towards a broader perspective and orientation, thus including CT and CS in their vision and priorities.

This resonates with the views Mary Dunphy Moloney (*Coderdojo*) and Pat Yongpradit (*Code.org*) voiced in the interviews. In particular, Dunphy Moloney pointed out that while coding activities help learners develop problem-solving and interpersonal skills, CT encompasses a broader set of skills. Yongpradit clarified that *Code.org*'s focus is on Computer Science for all, not only on coding.

Most of the initiatives described were founded in 2011 or later (except for *Bebras* in 2004 and *Computer Science Unplugged* in the 1990s). Most of the grassroots initiatives use a variety of terms on their official website. Three initiatives explicitly refer to CT in their mission statement (*Bebras, Code@SG Movement, Computer Science Unplugged*), four refer to coding/programming, four to Computer Science and two to Computing.

Overall, the grassroots initiatives have a high outreach, with numbers increasing every year. For instance, in 2015 *EU Code Week* organized 7600 events on learning to create with code in 46 countries (Europe, Africa, Asia, Australia, and the US) involving 570,000 people. The *Bebras* computing challenge involved more than 1,300,000 students from 38 countries, and tens of millions worldwide have tried the *Hour of Code*.

The majority of the grassroots initiatives also target formal education by developing course materials and organising teacher training. The *CS Unplugged* materials and *Code.org* courses are very popular in schools worldwide. Some grassroots initiatives focus on a specific target group. *CoderDojo*, for example, focuses on bringing together young people to participate in programming clubs, while *Computing At School* focuses on supporting teachers in England who are implementing the computing curriculum.

**Box 5.** Bebras

Bebras was founded in 2004 in Lithuania as a single-focus annual event. Since then it has developed into a multifunctional challenge that includes various activities, such as contest rounds, discussion on informatics topics, task-solving seminars, teacher workshops, and task developing events. In 2015, Bebras reached more than 1.3 million participants from 38 countries. The same year, the contest changed its name from "*Bebras contest on informatics and computer fluency*" to "*Bebras challenge on informatics and computational thinking*" (bebras.org). Bebras is the Lithuanian word for beaver, which is seen as a smart, hard-working and determined animal.

National Bebras challenges are held every year at the same time in autumn in all participating countries, usually in November. Each participant has 45 minutes to solve 18 tasks that focus on solving problems from a broad range of informatics topics, without requiring any programming skills. Different tasks are offered for five different age groups from 8 to 19 years. Teachers usually supervise the activity, which is performed at schools on computers. Several countries have established a second round of the Bebras challenge, usually at the end of January or beginning of February, which is dedicated to the best participants from the first round.

Both challenge rounds aim to trigger students' interest in Computer Science and to promote CT, embracing algorithmic, logical and operational skills, and based on informatics fundamentals. To this end, representatives of the Bebras countries aspire to wrap serious scientific problems and basic informatics concepts into playful tasks in an effort to attract and motivate students.

Grassroots initiatives that receive public funding also rely on backing from industry partners. Google, for instance, supports *Bebras*, *Code.org* and *Computer Science Unplugged*, while Microsoft supports *Code.org, CoderDojo* and *Computing At School* (CAS).

## Global Level

CoderDojo is a free after-school club, led by volunteers, for kids aged 7-17. Kids, parents, mentors, and others play with technology and learn to code. There are 1,100 plus active dojos in 65 countries in the world reaching 45,000 kids.

Launched in 2013, Code.org® is a non-profit dedicated to expanding access to Computer Science, and increasing the participation of women and underrepresented minorities. 263 million people have tried the *Hour of Code* world wide and 11 million students have used *Code Studio*.

Bebras is an international initiative whose goal is to promote Computer Science and CT mainly among teachers and pupils of all ages, but also among the public at large. Bebras organises easily accessible and highly motivating online challenges in many countries.

CS Unplugged is a collection of free learning activities that teach Computer Science through engaging games and puzzles. The initiative allows young students to dive head-first into Computer Science, without learning programming first.

Code Club builds a community of volunteers computer science experts who share a passion for digital making with children and teachers across the world. Code Club offers training for primary teachers Since 2012, Code Club has developed a growing network of after-school clubs.

Made With Code is a Google-sponsored initiative aimed at getting young women excited about learning to code and closing the gender gap in the tech industry. It provides resources, motivation, videos, and activities that enthuse girls about coding activities.

## European Level

EU Code Week is a grassroots movement run by volunteers who promote coding in their countries as Code Week Ambassadors. EU Commissioner Neelie Kroes launched the initiative in 2013. In 2015, more than 150,000 people participated in 4,200 coding events.

The European Coding Initiative, or 'all you need is {C<3DE}', brings together a wide array of stakeholders to promote coding and CT at all levels of education, as well as in informal settings.

The Barefoot Computing Programme supports UK primary school teachers in delivering the computing curriculum. It empowers teachers with the confidence, knowledge, skills and resources to teach computer science and help young people become 'computational thinkers'.

The mission of Computing At School is to provide leadership and strategic guidance to all those involved in Computing education in schools, with a significant focus on the Computer Science theme within the wider Computing curriculum.

Code it Like a Girl organizes workshops that familiarize women in Greece with coding. It aims to expand to all Greek cities and become the organization with the strongest impact on women in Greece.

Programamos is a Spanish non-profit organization whose objective is to promote the development of CT from an early age. It is set up as a social network that allows the exchange of best practices, resources, and ideas that promote programming, coding, and CT.

## Extra European Level

The Code@SG Movement, organised by the Infocomm Development Authority of Singapore (iDA), aims to seed coding and CT taught to students to build Singapore's national capability and to prepare for #SmartNation.

In Singapore, Computhink provides computer programming courses and holiday camps for kids aged 7 to 16. Launched in February 2015, its mission is to make CT and programming accessible to students of all ages.

Code for Change Myanmar is a new initiative designed to help the technology community in Myanmar use their skills to help tackle some of the country's pressing problems.

# 8. Conclusions

Currently, the integration of CT in formal and informal learning is a growing and very interesting trend in Europe and beyond. CT has become a buzzword that seems to promise the education of a new generation of children with a much deeper understanding of our digital world.

The information collected through this study points to a clear trend: the educational landscape is changing fast and we are now at a tipping point. Various initiatives, centred on CT in education - both top down and bottom up - are emerging worldwide. These all aim to provide young people with hands-on experience in CT-related activities, including programming/coding. In informal learning contexts, bottom-up initiatives have already moved ahead in providing experiences of this kind to many children in Europe and beyond.

However, in order to ensure equal opportunities and provide all children with the computer science skills they need to thrive in a digital economy, CT ought to be integrated in formal education. This will only deliver results if policy makers set out their vision and carefully define, plan and monitor their concrete implementation steps.

Setting specific goals is not only fundamental to informing concrete implementation choices, but also to getting relevant stakeholders on board. Several experts point out the importance of introducing children to CT concepts early on in school. Further evidence is needed to define the optimal introduction age. Currently, however, the general assumption is that CT-related skills should be developed from an early age and that even young children can grasp core CT concepts.

New comprehensive approaches are needed to cope with the complexity of cognitive processes related to CT. To help teachers assess CT skills, new tools and criteria are required. Support from national or transnational research programmes could prove instrumental in achieving this goal.

The introduction of CT in the curriculum is creating a strong demand for large-scale in-service continual professional development (CPD), as many teachers did not learn about CT in their initial education. It is of paramount importance that teacher and school staff should be provided with training opportunities that strongly focus on CT pedagogy and hands-on learning which can be easily transferred to the classroom. In addition, policy actions could also include peer exchanges and community building to enable the sharing of best practices among teachers.

## 8.1 Implications for policy and practice

The policy implications outlined in this section suggest ways policy makers can further ease the introduction of CT in compulsory education. The recommendations cover different phases, from planning to implementation, and are all strongly interrelated.

At the policy level, it may be important for national/regional policy makers to focus on four important areas: *consolidated CT understanding; comprehensive integration; systemic rollout;* and *policy support*. Figure 9 depicts these four areas and related set of recommendations.

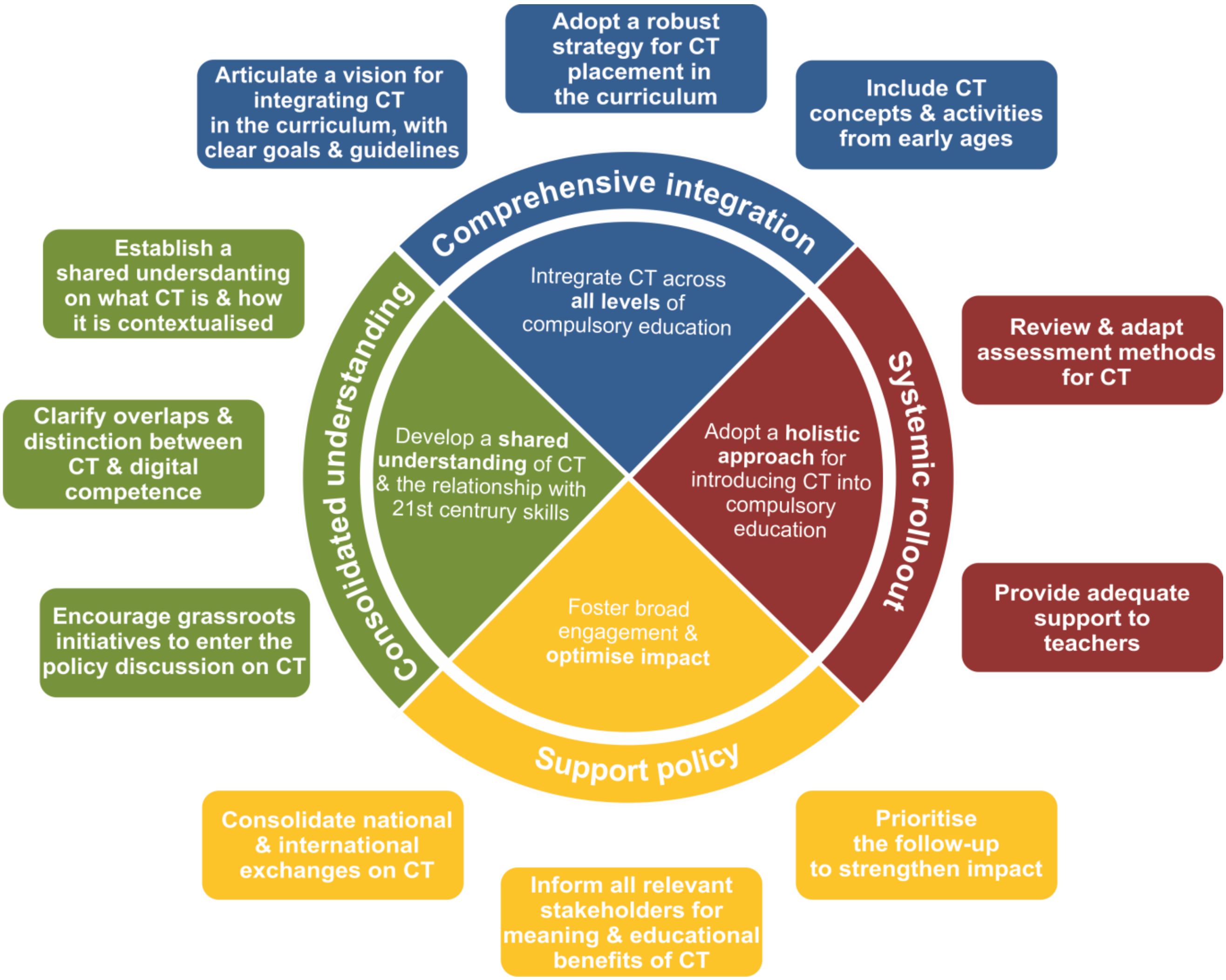



**Figure 9**. Introducing CT into compulsory education: implications for policy and practice

### FOCUS: CONSOLIDATED UNDERSTANDING

**Establish a shared understanding of what CT is and how this is contextualised**

This plurality of terms can cause confusion, especially when different terms like CT and algorithmic thinking are used to refer to the same core concept. Engagement with relevant experts may help policy makers within Ministries of Education to develop a common understanding of terms.

To this end, exchange across countries might be beneficial. Consolidation of shared terminology may facilitate the process of curriculum integration, while at the same time respecting teachers' freedom to introduce CT in a way that is suited to their specific school context. One example of this approach is the Czech Strategy of Digital Education, which already provides a description of the area of CT.

**Clarify the overlaps and distinctions between CT and digital competence**

In the academic literature, little or no evidence has been collected on the relationship between digital competence and CT. Some experts are adamant that they are two different concepts. However, curricula and policy documents tend to treat them as related topics. Evidence is emerging in some countries of a clear shift away from a focus on students' practical ICT skills towards an approach that focuses on underlying computer and design principles, while putting students in the role of creators. Some current conceptualisations of Digital Competence (e.g. DigComp) point in this direction, indicating that digital competence is much wider than practical ICT skills. As the discussion is more political than scientific or conceptual, it may be advisable for working groups within Ministries of Education to discuss the convergences between digital competences and CT. As part of such discussions, it may be worth addressing specific questions such as whether CT fosters students' digital skills/competence.

**Encourage grassroots initiatives to enter the policy discussion on CT**

This study reports on several recently launched initiatives that have rapidly reached global impact in terms of informal learning. The experience gained and lessons learned from these initiatives can provide valuable input for the integration of CT in formal education, provided that they are tailored to the specific formal education contexts. One strategy for creating fruitful cooperation and synergies between formal and informal education could be for grassroots initiatives to deliver expertise and to support teacher training; by the same token, formal education could refer particularly gifted and motivated students to these initiatives as further extra-curricular learning opportunities. Another potential synergy is that integrating the teaching of programming in formal education is a platform for introducing the hitherto uninitiated (especially girls) to programming and a springboard for personal projects that are pursued outside formal settings and are highly motivating. For example, in Finland the establishment of afternoon code clubs for children has gone hand in hand with the integration of programming in the new core curriculum, which commenced implementation in autumn 2016.

## FOCUS: COMPREHENSIVE INTEGRATION

**Articulate a vision for integrating CT in compulsory education, with clear goals**

Defining a clear vision for the integration of CT in compulsory education is crucial. It may be useful to begin by setting specific goals that inform concrete implementation choices internally and then bring stakeholders on board at a later stage. The survey of ministries and expert interviews indicate that while the main reasons for introducing CT are common to many countries, the specific focus can differ. Finland, for instance, emphasises programming as a means to get more students interested in learning maths. The experts highlighted that, although CT initiatives are on the rise in many European countries and beyond, few countries offer CT classes to all learners as a compulsory element of their education. This choice should also be considered in the light of the overall vision and specific goals of integrating CT in formal education.

**Adopt a robust strategy for CT placement in the curriculum**

As CT involves far more than offering a few hours of coding, placement in the curriculum calls for a robust strategy that accounts for the wide range of factors involved. A key consideration is the extent to which CT is allocated across the full spectrum of subject area studies and, also, in multi-disciplinary and inter-disciplinary contexts. This also implies careful planning and selection of pedagogical approaches, as well as tools and assessment strategies in order to enhance students' understanding of core CT concepts incrementally over several school years.

**Include CT concepts and activities from early ages**

Several experts stress the importance of introducing CT concepts to children early on in school. The general assumption is that essential related competences need to be developed from an early age and that it's possible for young children to grasp the core concepts of CT. A number of visual programming tools suitable for young children are widely available. While CT is still most commonly integrated in secondary education, more and more countries are now integrating it at primary level as well.

## FOCUS: SYSTEMIC ROLLOUT

**Review and adapt innovative assessment methods**

The importance of assessment for full and effective integration of CT in education is clearly highlighted in the literature. In most cases, the strategy adopted for CT assessment is to analyse the artefacts (e.g. games or models) that students develop as indications of their CT abilities. Other strategies include multiple-choice tests, attendant rubrics to assess CT skills, or getting students to modify the code of an existing program so as to accomplish specific objectives. Design-based approaches, such as programming interactive media, are also emerging as key elements of assessments systems. However, few signs are emerging of new, comprehensive approaches that encompass the complexity of the cognitive processes in place with CT. Hence the need to define new tools and criteria to help teachers assess CT skills, in particular as part of a cross-curricular approach. Further policy actions are required to investigate, develop and pilot new assessment methods.

**Provide adequate support to teachers**

There is broad consensus emerging, in particular from the grey literature and expert interviews, that the introduction of CT in compulsory education requires support measures for teachers. Curriculum integration creates demand for large-scale in-service continuous professional development, as many teachers did not learn about CT in their initial training. Moreover, teaching CT may require new pedagogical approaches that put students at the centre of the learning process. Policy makers may want to consider the provision of training opportunities that have a strong focus on pedagogy and that involve hands-on learning that is easy to transfer to the classroom. In addition, some emphasis could also be placed on fostering peer exchange and community building, enabling the sharing of good practice among teachers.

## FOCUS: SUPPORT POLICY

### Consolidate national and international exchanges on CT among policy makers, grassroots initiatives, research centres and other stakeholders

*"When something is working already elsewhere, it is better to copy it and tailor it to your own purpose than trying to do everything from the beginning yourself. Co-operation is key, and also the sharing of failures", Leo Pahkin (FNBE, Finland).*

Exchange with policy makers from other countries, especially those that are more advanced in the area, can yield extremely valuable insights, especially as the reasons for integrating CT are similar across countries. Furthermore, by sharing their initial results, policy makers in countries that have already integrated CT may be inspired to rethink and refine their next steps, while taking account of their own national and local context. Such exchanges among policy makers, but also with the representatives of grassroots initiatives and with research centres, may generate further thinking on common challenges like the assessment of CT or successful models of teacher training.

### Inform all relevant stakeholders of the meaning, conceptualisations and educational benefits of CT

At the stage of actual implementation in the curriculum, it is highly advisable to inform stakeholders not directly involved in curricula development about what CT is, what it entails, and what educational advantages it offers. It is especially important for teachers and educators to have a solid understanding of what CT is and how to teach it. The relevance of CT also needs to be clear to parents and other stakeholders. For instance, as part of its implementation of the national Strategy of Digital Education, the Czech MOE is planning an information campaign to introduce the concept of CT to teachers and school heads, but also to parents and the general public.

### Prioritise the follow-up to strengthen impact

Given the need to adopt a holistic approach for effective integration of CT into compulsory education, it follows that a wide-angle monitoring and analysis strategy is required to measure the impact and sustainability of implemented actions. Undoubtedly, these measures need to be calibrated against clear, predefined goals specified at policy and strategy levels. This (qualitative) benchmarking process underpins efforts to support the exchange of experiences and best practices, both at local/national level and beyond.

To conclude, while CT is a very promising concept that can help prepare children for future challenges in an increasingly digital society with digital job markets, the proof of concept has yet to be delivered. If CT is to earn a place in the curriculum over the coming decades, results in the coming years will need to demonstrate that teaching CT has an actual impact on children's learning and skills. Hence, as more tangible results on the concrete implementation and pedagogical choices become available in many countries, the exchange of experience and lessons learned at both European and international levels will become crucial. Rigorous research on specific aspects such as assessment methods and transfer of knowledge will be key to the successful implementation of CT in formal education.

# References

Académie des Sciences. (2013). *L'enseignement de l'informatique en France - Il est urgent de ne plus attendre.*

Allan, W., Coulter, B., Denner, J., Erickson, J., Lee, I., Malyn-Smith, J., & Martin, F. (2010). Computational Thinking for Youth. ITEST Small Working Group on Computational Thinking.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 Computational Thinking Curriculum Framework- Implications for Teacher Knowledge. *Educational Technology & Society*, *19*(3), 47–57.

Armoni, M. (2010). On Teaching Topics in Computer Science Theory. *ACM Inroads*, *1*(1), 21–22.

Armoni, M., & Gal-Ezer, J. (2014). Early Computing Education: Why? What? When? Who? *ACM Inroads*, *5*(4), 54–59.

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. Robotics and Autonomous Systems, 75, Part B, 661–670. https://doi.org/10.1016/j.robot.2015.10.008

Barr, D., Harrison, J., & Conery, L. (2011). Computational Thinking: A Digital Age Skill for Everyone. *Learning & Leading with Technology*, *38*(6), 20–23.

Barr, V., & Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads*, *2*(1), 48–54.

Bell, T., Rosamond, F., & Casey, N. (2012). Computer Science Unplugged and Related Projects in Math and Computer Science Popularization. In H. L. Bodlaender, R. Downey, F. V. Fomin, & D. Marx (Eds.), *The Multivariate Algorithmic Revolution and Beyond* (pp. 398–456). Springer Berlin Heidelberg.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *AERA2012 - annual meeting of the American Educational Research Association*. Vancouver, Canada.

Charlton, P., & Luckin, R. (2012). *Time to re-load? Computational Thinking and Computer Science in Schools* (Briefing 2, 27 April). The London Knowledge Lab.

Council of the European Union, European Commission. (2015). 2015 Joint Report of the Council and the Commission on the implementation of the strategic framework for European cooperation in education and training (ET 2020). New priorities for European cooperation in education and training.

Creative Learning Exchange. (2015). Using System Dynamics and Systems Thinking (SD/ST) Tools and Learning Strategies to Build Science, Technology, Engineering, and Math Excellence.

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational thinking A guide for teachers. Computing at School.

CSTA Standards Task Force. (2016). *[Interim] CSTA K-12 Computer Science Standards*. New York: CSTA.

CSTA & ISTE. (2009). Operational definition of Computational Thinking in K-12 education. Retrieved from https://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf

Curzon, P., McOwan, P. W., Plant, N., & Meagher, L. R. (2014). Introducing Teachers to Computational Thinking Using Unplugged Storytelling. In Proceedings of the 9th Workshop in Primary and Secondary Computing Education (pp. 89–92). New York, NY, USA: ACM. https://doi.org/10.1145/2670757.2670767

Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn coding? In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education* (pp. 60–69). Berlin, Germany: ACM.

McCormack, A. (2014). The e-Skills Manifesto. Brussels.

European Commission. (2016). COM (2016) 581 final. *Strengthening European Investments for jobs and growth: Towards a second phase of the European Fund for Strategic Investments and a new European External Investment Plan*. European Commission

European Commission. (2016). COM (2016) 381/2. A New Skills Agenda for Europe. Working together to strengthen human capital, employability and competitiveness {SWD (2016) 195}. European Commission.

European Parliament and The Council of European Union (2006). Key Competences Recommendation 2006/962/EC of the European Parliament and of the Council of 18 December 2006 on key competences for lifelong learning [Official Journal L 394 of 30.12.2006].

Ferrari, A. (2013). DIGCOMP: A Framework for Developing and Understanding Digital Competence in Europe. Yves Punie and Barbara N. Brečko (Eds). JRC Scientific and Policy Reports. EUR JRC83167 EN. Luxembourg: Publications Office of the European Union. Retrieved from: http://ipts.jrc.ec.europa.eu/publications/pub.cfm?id=6359

Gander, W., Petit, A., Berry, G., Demo, B., Vahrenhold, J., McGettrick, A., … Drechsler, M. (2013). Informatics education: Europe cannot afford to miss the boat (p. 21). Joint Informatics Europe & ACM Europe Working Group on Informatics Education.

Gretter, S., & Yadav, A. (2016). Computational Thinking and Media & Information Literacy: An Integrated Approach to Teaching Twenty-First Century Skills. *TechTrends*, 1–7.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43.

Grover, S., Cooper, S., & Pea, R. (2014). Assessing Computational Learning in K-12. In *Proceedings of the 2014 Conference on Innovation; Technology in Computer Science Education* (pp. 57–62). ACM USA.

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. Computer Science Education, 25(2), 199–237.

Informatics Europe and ACM Europe (2013). Informatics education: Europe cannot afford to miss the boat. Retrieved from http://www.informatics-europe.org/images/documents/informatics-education- europe-report.pdf

ISTE & CSTA. (2011). Operational Definition of Computational Thinking for K–12 Education.

Ioannidou, A., Bennett, V., Repenning, A., Koh, K. H., & Basawapatna, A. (2011). Computational Thinking Patterns. In AERA Conference Proceedings 2011. Retrieved from http://eric.ed.gov/?id=ED520742

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., … Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37.

Margolis, J. (2008). Stuck in the Shallow End. The MIT Press. Retrieved from https://mitpress.mit.edu/books/stuck-shallow-end

Moreno-León, J., Robles, G., & Román-González. (2015). Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking. RED. Revista de Educación a Distancia, 15(46). Retrieved from http://www.um.es/ead/red/46/moreno_robles.pdf

Mouza, C., Pollock, L., Pusecker, K., Guidry, K., Yeh, C.-Y., Atlas, J., & Harvey, T. (2016). Implementation and Outcomes of a Three-Pronged Approach to Professional Development for CS Principles. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 66–71). New York, NY, USA: ACM.

National Research Council. (2010). *Report of a Workshop on The Scope and Nature of Computational Thinking*. Washington, D.C.: National Academies Press.

National Research Council. (2011). *Report of a Workshop on the Pedagogical Aspects of Computational Thinking*. Washington, D.C.: National Academies Press.

Papert, S. (1980). Mindstorms: Children, Computers, and Powerful Ideas. New York, NY, USA: Basic Books, Inc.

Repenning, A., Webb, D. C., Koh, K. H., Nickerson, H., Miller, S. B., Brand, C., … Repenning, N. (2015). Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools Through Game Design and Simulation Creation. ACM Transactions on Computing Education, 15(2), 11:1–31.

Ryoo, J., Goode, J., Margolis, J. (2016): It takes a village: supporting inquiry- and equity-oriented computer science pedagogy through a professional learning community, Computer Science Education, DOI: 10.1080/08993408.2015.1130952

Saari, E. M., Blanchfield, P., & Hopkins, G. (2015). Computational Thinking: A Tool to Motivate Understanding in Elementary School Teachers. In S. Zvacek, M. T. Restivo, J. Uhomoibhi, & M. Helfert (Eds.), *Computer Supported Education* (pp. 348–364). Springer International Publishing.

Sax, L. J. (2012). Examining the Underrepresentation of Women in STEM Fields: Early Findings from the Field of Computer Science (UCLA Center for the Study of Women). University of California, Los Angeles. Retrieved from https://escholarship.org/uc/item/84j9s1v1

Selby, C. C. (2014). *How Can the Teaching of Programming Be Used to Enhance Computational Thinking Skills?* (2). University of Southampthon, Southampthon, UK.

Selby, C. C., & Woollard, J. (2013). Computational Thinking: The Developing Definition. University of Southampton (E-prints).

Sentance, S., & Humphreys, S. (2015). Online vs Face-To-Face Engagement of Computing Teachers for their Professional Development Needs. In A. Brodnik & J. Vahrenhold (Eds.), *Informatics in Schools. Curricula, Competences, and Competitions* (Vol. 9378, pp. 69–81). Cham: Springer Publishing.

Snodgrass, M. R., Israel, M., & Reese, G. C. (2016). Instructional supports for students with disabilities in K-5 computing: Findings from a cross-case analysis. Computers & Education, 100, 1–17. https://doi.org/10.1016/j.compedu.2016.04.011

The Royal Society. (2012). *Shut down or restart? The way forward for computing in UK schools*.

Vuorikari, R., Punie, Y., Carretero Gomez S., Van den Brande, G. (2016). DigComp 2.0: The Digital Competence Framework for Citizens. Update Phase 1: The Conceptual Reference Model. Luxembourg Publication Office of the European Union. EUR 27948 EN.

Webb, M., Davis, N., Katz, Y. J., Reynolds, N., & Syslo, M. M. (2015). Towards deeper understanding of the roles of CS/ Informatics in the curriculum. In A. Brodnik & C. Lewin (Eds.), IFIP TC3 Working Conference "A New Culture of Learning: Computing and next Generations." Vilnius University, Lithuania. Retrieved from http://www.ifip2015.mii.vu.lt

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2016). Computer science in K-12 school curricula of the 2lst century: Why, what and when? Education and Information Technologies, 1–24. https://doi.org/10.1007/s10639-016-9493-x

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2015). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 1–21.

Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010). *Running On Empty: The Failure to Teach K–12 Computer Science in the Digital Age*. ACM&CSTA.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717–3725.

Wing, J. M. (2011). Research Notebook: Computational Thinking-What and Why? *The Link*.

Woollard, J. (2016). CT Driving Computing Curriculum in England. *CSTA Voice*, *12*(1), 4–5.

Yadav, A., Burkhart, D., Moix, D., Snow, E., Bandaru, P., & Clayborn, L. (2015). *Sowing the Seeds: A Landscape Study on Assessment in Secondary Computer Science Education*.

Yadav, A. (2014). Computational Thinking in K-12 Education. [PowerPoint slides], Purdue University, Indiana.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational Thinking in Elementary and Secondary Teacher Education. *ACM Transactions on Computing Education*, *14*(1), 5:1–5:16.

Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital (Computational Thinking: A New Digital Literacy). *Revista de Educación a Distancia*, *46*(3).

## List of Boxes



## List of Figures



## List of Tables

# Annex 1: Ministries of Education contributed to the survey

| Country | Organisation |
|---|---|
| **Austria** | BMBF, IT-Didactic and Digital Media |
| **Czech Republic** | The Centre for International Cooperation in Education (DZS) |
| **Denmark** | Ministry for children, education, and gender equality |
| **Estonia** | The Information Technology Foundation for Education (HITSA) |
| **Finland** | Finnish National Board of Education (FNBE) |
| **France** | Ministry for Education, Higher Education and Research |
| **Greece** | Greek Institute of Education Policy and the Directorates of the MOE for primary, secondary and VET education[14] |
| **Hungary** | Educational Authority |
| **Israel** | Ministry of Education |
| **Italy** | Ministry of Education, Dipartimento per la programmazione - direzione generale per l'edilizia i fondi strutturali e l'innovazione digitale |
| **Lithuania** | Education Development Centre |
| **Malta** | Ministry for Education and Employment |
| **Norway** | Norwegian Centre for ICT in Education |
| **Poland** | Ministry of National Education |
| **Portugal** | Direção-Geral da Educação (DGE) |
| **Spain** | MOE - Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado |
| **Sweden** | Skolverket (Swedish National Agency for Education) |
| **Switzerland – German speaking schools** | Deutschschweizer Erziehungsdirektoren-Konferenz, educa.ch |
| **Switzerland – French speaking schools** | Conférence intercantonale de l'instruction publique de la Suisse romande et du Tessin |
| **Turkey** | Ministry of National Education, Directorate General for Innovation and Educational Technologies |

[14] As noted already (see footnote 2, p. 14), the Greek Institute of Education Policy and the Directorates of the MOE for primary, secondary and VET education did not reply to the survey questionnaire but provided CompuThink team with general information on the status of CT in the Greek education system.

# Annex 2: Experts contributed to the semi-structured interviews

| SURNAME | Name | Affiliation | Country | Role |
|---|---|---|---|---|
| **DUNPHY MOLONEY** | Mary | CoderDojo | Ireland | Expert |
| **GAL-EZER** | Judith | Open University of Israel | Israel | Researcher |
| **GREČNEROVÁ** | Barbora | Dům zahraniční spolupráce (DZS) | Czech Republic | Policy maker |
| **KRUPA** | Janusz | Ministry of National Education | Poland | Policy maker |
| **LEE** | Irene | Chair of CSTA CT task force and MIT Researcher | United States | Researcher |
| **LEPELTAK** | Jan | CEPIS and director of LearningFocus | The Netherlands | Expert |
| **PAHKIN** | Leo | Finnish national board of Education | Finland | Expert |
| **PEYTON JONES** | Simon | Microsoft Research in Cambridge, England | United Kingdom | Researcher |
| **RESNICK** | Mitchel | MIT Media Lab | United States | Researcher |
| **SANDERS** | Ruth | Special education Teacher at Ysgol Hendrefelin School in Wales | United Kingdom | Teacher |
| **SYSLO** | Maciej M. | Ministry of National Education, Universities of Toruń and Wrocław | Poland | Researcher |
| **URSCHITZ** | Tullia | Math and science teacher and Italian Scientix Ambassador | Italy | Teacher |
| **VITIKKA** | Erja | Finnish national board of Education | Finland | Stakeholder |
| **VOOGT** | Joke | University of Amsterdam | The Netherlands | Researcher |
| **YONGPRADIT** | Pat | Chief Academic Officer at Code.org | United States | Expert |

## Annex 3: Type of information sources per Country

| Country | Type of information sources | Selected references from the literature |
|---|---|---|
| **Austria** | - Survey with MOE<br>- Desk research | [1] Bundesministerium für Bildung und Frauen. (2004). AHS Obserstufe Lehrplan Informatik. Retrieved from https://www.bmb.gv.at/schulen/unterricht/lp/lp_neu_ahs_14_11866.pdf?5h6vve<br>[2] Sabitzer, B., Antonitsch, P. K., & Pasterk, S. (2014). Informatics Concepts for Primary Education: Preparing Children For Computational Thinking. In Proceeding of WiPSCE'14. Berlin, Germany: ACM Press. |
| **Belgium** | Desk research | [3] VanDaele, A. (2014, November 24). Informatics in Flemish schools (Belgium) CECE Blog [Blog post]. Retrieved from http://ceceblog.netzverwaltung.info/?p=42 |
| **Croatia** | Desk research | [4] Kralj, L. (2016). New Informatics curriculum - Croatian tradition with world trends. In 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 760–763). |
| **Cyprus** | Desk research | [5] Fluck, A., Webb, M., Cox, M., Angeli, C., Malyn-Smith, J., Voogt, J., & Zagami, J. (2016). Arguing for Computer Science in the School Curriculum.pdf. Educational Technology & Society, 19(3), 38–46.<br>[6] Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2016). Computer science in K-12 school curricula of the 2lst century: Why, what and when? Education and Information Technologies, 1–24.<br>[7] Balanskat, A., & Engelhardt, K. (2015). Computing our future. Computer programming and coding. (p. 87). Bruxelles: European Schoolnet. |
| **Czech Republic** | - Survey with MOE<br>- Desk research<br>- Interview | [8] Czech Ministry of Education, Youth and Sports. (2014). Strategy for Education Policy of the Czech Republic until 2020 (Strategy Paper). Retrieved from http://www.vzdelavani2020.cz/images_obsah/dokumenty/strategy_web_en.pdf<br>[9] Grečnerová, B. (2015). Czech Republic. Country Report on ICT in Education (Country Report). European Schoolnet. |
| **Denmark** | - Survey with MOE<br>- Desk research | [10] Caspersen, M. E., & Nowack, P. (2013). Computational thinking and practice: a generic approach to computing in Danish high schools. In Proceedings of the Fifteenth Australasian Computing Education Conference (Vol. 136, pp. 137–143). Australian Computer Society, Inc. Retrieved from http://www.cs.au.dk/~mec/publications/conference/41--ace2013.pdf |
| **England (UK)** | - Desk research<br>- Interview | [11] Department for Education. (2013, September 11). National curriculum in England: computing programmes of study.<br>[12] Boylan, M., & Willis, B. (2015). Independent Study of Computing At School Master Teacher programme. Sheffield Hallam University. |
| **Estonia** | - Survey of MOE<br>- Desk research | [13] Ministry of Education and Research. (2015). Estonian Lifelong Learning Strategy 2020. Retrieved from https://www.hm.ee/sites/default/files/estonian_lifelong_strategy.pdf<br>[14] ProgeTiger program: http://www.hitsa.ee/it-education/educational-programmes/progetiger |

| Country | Type of information sources | Selected references from the literature |
|---|---|---|
| **Finland** | - Survey of MOE<br>- Desk research<br>- Interviews | [15] Finnish National Board of Education (FNBE) (2016). National core curriculum for basic education 2014 (English version). Finnish National Board of Education 2016. Retrieved from http://www.oph.fi/english/education_development/current_reforms/curriculum_reform_2016<br>[16] Halinen, I. (2016). Curriculum reform in Finland. [PowerPoint slides]. Retrieved from http://www.oph.fi/english/education_development/current_reforms/curriculum_reform_2016<br>[17] Halinen, I. (2015, March 23). What is going on in Finland? – Curriculum Reform 2016 [Blog post]. Retrieved from http://oph.fi/english/current_issues/101/0/what_is_going_on_in_finland_curriculum_reform_2016 |
| **France** | - Survey of MOE<br>- Desk research | [18] French Government. (2015). Socle commun de connaissances, de compétences et de culture. Retrieved from http://www.education.gouv.fr/pid25535/bulletin_officiel.html?cid_bo=87834<br>[19] French Government. (2015). Programmes d'enseignement du cycle des apprentissages fondamentaux (cycle 2), du cycle de consolidation (cycle 3) et du cycle des approfondissements (cycle 4). Retrieved from http://www.education.gouv.fr/pid285/bulletin_officiel.html?pid_bo=33400<br>[20] Pène, S., Abiteboul, S., Balagué, C., Blecher, L., Bloch-Pujo, N., Briand, M., … Vallée, B. (2014). Jules Ferry 3.0, Bâtir une école créative et juste dans un monde numérique. Conseil National du Numérique. Retrieved from http://www.cnnumerique.fr/education-2/ |
| **Germany** | - Desk research | [21] Knobelsdorf, M., Magenheim, J., Brinda, T., Engbring, D., Humbert, L., Pasternak, A., … Vahrenhold, J. (2015). Computer Science Education in North-Rhine Westphalia, Germany—A Case Study. ACM Transactions on Computing Education, 15(2), 1–22.<br>[22] Hubwieser, P. (2012). Computer Science Education in Secondary Schools – The Introduction of a New Compulsory Subject. *Transactions of Computing Education*, 12(4), 16:1–16:41. |
| **Greece** | - Note from IEP and MOE<br>- Desk research | [23] Committee of continuous Educational affairs of the Greek parliament (2016). National and social dialogue for education. Observations, suggestions, and implementation timeframe. Retrieved from https://www.minedu.gov.gr/publications/docs2016/morfotikwn_porisma.pdf |
| **Hungary** | - Survey of MOE<br>- Desk research | [24] ISMAN, A., Biró, P., Csernoch, M., Máth, J., & Abari, K. (2015). International Educational Technology Conference, IETC 2014, 3-5 September 2014, Chicago, IL, USA Measuring the Level of Algorithmic Skills at the End of Secondary Education in Hungary. In Procedia - Social and Behavioral Sciences, 176, pp. 876–883.<br>[25] Főző, A., & Csordás, I. (2015). Country Report on ICT in Education - Hungary. European Schoolnet.<br>[26] Zsakó, L., Szlávi, P., Zsakó, L., & Szlávi, P. (2012). ICT Competences: Algorithmic Thinking. Acta Didactica Napocensia, 5(2), 49–58.<br>[27] Hungary Governament (October 13, 2016). *Digital Education Strategy*. Retrieved from http://www.kormany.hu/hu/miniszterelnoki-kabinetiroda/digitalis-jolet-program/strategiak |

| Country | Type of information sources | Selected references from the literature |
|---|---|---|
| **Ireland** | - Desk research | [28] Department of Education and Skills, I. (2015). Digital Strategy for school 2015-2020. Enhancing teaching learning and assessment. Department of Education. Retrieved from https://www.education.ie/en/Publications/Policy-Reports/Digital-Strategy-for-Schools-2015-2020.pdf<br>[29] CEPIS. (2013). The Irish Computer Society's Computing Curriculum Rolled Out in 70 Irish Schools. Retrieved from http://www.ics-skills.ie/education/curriculum-computing.php |
| **Israel** | - Survey of MOE<br>- Desk research<br>- Interview | [30] Gal-Ezer, J., & Stephenson, C. (2014). A Tale of Two Countries: Successes and Challenges in K-12 Computer Science Education in Israel and the United States. ACM Transactions on Computing Education, 12(2), 8:1–8:18.<br>[31] Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2016). Computer science in K-12 school curricula of the 2lst century: Why, what and when? Education and Information Technologies, 1–24.<br>[32] Bargury, I. Z., Muller, O., Haberman, B., Zohar, D., Cohen, A., Levy, D., & Hotoveli, R. (2012). Implementing a new Computer Science Curriculum for middle school in Israel. In Frontiers in Education Conference (FIE), 2012 (pp. 1–6).<br>[33] Armoni, M., & Gal-Ezer, J. (2014). High school computer science education paves the way for higher education: the Israeli case. Computer Science Education, 24(2–3), 101–122. |
| **Italy** | - Survey of MOE<br>- Desk research | [34] Ministry of Education and Research of Italy. (2015). Piano Nazionale Scuola Digitale. MIUR. Retrieved from http://www.istruzione.it/scuola_digitale/allegati/Materiali/pnsd-layout-30.10-WEB.pdf<br>[35] Bellettini, C., Lonati, V., Malchiodi, D., Monga, M., Morpurgo, A., Torelli, L., & Zecca, L. (2014). Informatics Education in Italian Secondary Schools. ACM Transactions on Computing Education, 14(2). |
| **Lithuania** | - Survey of MOE<br>- Desk research | [36] Dagiene, V., & Stupuriene, G. (2016). Informatics Concepts and Computational Thinking in K-12 Education: A Lithuanian Perspective. Journal of Information Processing, 24(4), 732–739. http://doi.org/10.2197/ipsjjip.24.73 |
| **Malta** | - Survey of MOE<br>- Desk research | [37] Ministry of Education and Employment of Malta. (2012). A National Curriculum Framework for all. Retrieved from http://curriculum.gov.mt/en/Resources/The-NCF/Documents/NCF.pdf<br>[38] Catania, J. (2014). Computing as a Core Entitlement Framework. Introducing Computing (Digital Literacy) as a core entitlement for all students. Department of eLearning & Dev. of Curriculum - DQSE- MEDE Malta. https://www.dropbox.com/s/1mw6nrlzqkltz0u/Computing%20as%20a%20Core%20Entitlement.pdf?dl=0 |
| **Norway** | - Survey of MOE<br>- Desk research | [39] Norwegian Centre for ICT in Education. (2016). Coding and programming in schools (White Paper). |
| **Poland** | - Survey of MOE<br>- Desk research | [40] Sysło, M. M., & Kwiatkowska, A. B. (2015). Introducing a New Computer Science Curriculum for All School Levels in Poland. In A. Brodnik & J. Vahrenhold (Eds.), Informatics in Schools. Curricula, Competences, and Competitions: 8th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2015, Ljubljana, |

| Country | Type of information sources | Selected references from the literature |
|---|---|---|
| | | Slovenia, September 28 - October 1, 2015, Proceedings (pp. 141–154). Cham: Springer International Publishing.<br>[41] Webb, M., Davis, N., Katz, Y. J., Reynolds, N., & Syslo, M. M. (2015). Towards deeper understanding of the roles of CS/ Informatics in the curriculum. In A. Brodnik & C. Lewin (Eds.), IFIP TC3 Working Conference 'A New Culture of Learning: Computing and next Generations'. Vilnius University, Lithuania.<br>[42] Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2016). Computer science in K-12 school curricula of the 2lst century: Why, what and when? Education and Information Technologies, 1–24. http://doi.org/10.1007/s10639-016-9493-x |
| **Portugal** | - Survey of MOE<br>- Desk research | [43] Minitério da Educação e Ciência. (2015). Projeto "Iniciação à Programação no 1.º Ciclo do Ensino Básico". Retrieved from http://www.erte.dge.mec.pt/iniciacao-programacao-no-1o-ciclo-do-ensino-basico<br>[44] Ramos, J. L., & Espadeiro, R. G. (2015). Pensamento computacional na escola e práticas de avaliação das aprendizagens. Uma revisão sistemática da literatura. In Atas do Challenges 2015. Universidade do Minho.<br>[45] Moreira, F., & Ferreira, M. J. (2015). Teaching Algorithms Profile-Oriented: a proposed methodology to elementary school. EDULEARN15 Proceedings, 309–317. |
| **Scotland (UK)** | - Desk research | [46] Scott, J., & Bundy, A. (2015). Creating a new generation of computational thinkers. Communications of the ACM, 58(12), 37–40.<br>[47] Scott, J. (2013). The royal society of Edinburgh/British computer society computer science exemplification project. In Proceedings of the 18th ACM conference on Innovation and technology in computer science education (pp. 315–315). ACM.<br>[48] The Scottish Government (2016, September 6). Enhancing Learning and Teaching Through the Use of Digital Technology. ISBN: 978-1-78652-473-7. Retrieved from http://www.gov.scot/Resource/0050/00505855.pdf |
| **Slovakia** | - Desk research | [49] Kalas, I. (2015). Programming at pre-primary and primary levels: the pipeline can start that early. KEYCIT 2014: Key Competencies in Informatics and ICT, 7, 29.<br>[50] Gujberova, M., & Kalas, I. (2013). Designing Productive Gradations of Tasks in Primary Programming Education. In Proceedings of the 8th Workshop in Primary and Secondary Computing Education (pp. 108–117). New York, NY, USA: ACM. |
| **Spain*** | - Survey of MOE<br>- Desk research | [51] Autonomous community of Andalucía: subject in the last year of Primary Education (11-12 year-old students): "CULTURA Y PRÁCTICA DIGITAL" ("Digital Culture and Practice". See introduction and blocs of contents 1,2 and 3) http://www.juntadeandalucia.es/educacion/descargasrecursos/curriculo-primaria/culturaydigital.html<br>[52] Atonomous community of Canarias: subject in the first two years of Compulsory Secondary Education (12-14 year-old students): PRÁCTICAS COMUNICATIVAS Y CREATIVAS ("Communicative and creative practices". See the introduction) http://www.gobiernodecanarias.org/opencmsweb/export/sites/educacion/web/_galerias/descargas/Secundaria/Orden |

| Country | Type of information sources | Selected references from the literature |
|---|---|---|
| | | acion_curriculo/borrador_nuevo_curriculo/nuevas/practicas_comunicativas_creativas_v_13_octubre.pdf<br>[53] Autonomous community of Cantabria: subject in the third year of Compulsory Secondary Education (14-15 year-old students): SISTEMAS DE CONTROL Y ROBÓTICA https://boc.cantabria.es/boces/verAnuncioAction.do?idAnuBlob=290641<br>[54] Autonomous community of Castilla – La Mancha: subject in the first year of Compulsory Secondary Education (12-13 years old): TECNOLOGÍA CREATIVA, pages 672-677. Also the subject in the fourth year of Compulsory Secondary Education (15-16 years old): TECNOLOGÍA ROBÓTICA, pages 678-683. http://www.educa.jccm.es/es/sistema-educativo/curriculo-lomce-horarios-bachillerato.ficheros/179327-Decreto%2040-2015_curr%C3%ADculo%20ESO-BTO.pdf<br>[55] Autonomous community of Castilla y León: the optional subject in the fourth year of Compulsory Secondary Education (15-16 years old): PROGRAMACIÓN INFORMÁTICA. The minimum teachings have not yet been established, though the subject is already mentioned in the official curriculum for this autonomous community (page 9). https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&cad=rja&uact=8&ved=0ahUKEwjxi9-c3YvMAhUEWRQKHS0tB_IQFggrMAQ&url=http%3A%2F%2Fwww.educa.jcyl.es%2Fes%2Finformacion%2Fsistema-educativo%2Feducacion-secundaria-obligatoria%2Fnueva-regulacion-educacion-secundaria-obligatoria.ficheros%2F549414-Orden%2520Curr%25C3%25ADculo%2520ESO.pdf&usg=AFQjCNF2g2i4NibGVV9-0qbTtkgQ0-xtZg&sig2=mrlhGztRo69I7wgIqOO8_w&bvm=bv.119408272,d.d24<br>[56] Autonomous community of Región de Murcia: the optional subject in the second year of Compulsory Secondary Education (13-14 years old): ROBÓTICA, pages 753-757. http://www.borm.es/borm/documento?obj=anu&id=735576<br>[57] Autonomous community of La Rioja: The subject TECNOLOGÍAS DE LA INFORMACIÓN Y DE LA COMUNICACIÓN in the first (12-13 years old) and fourth year (15-16 years old) of Compulsory secondary Education. See pages 321-325. http://ias1.larioja.org/boletin/Bor_Boletin_visor_Servlet?referencia=2386883-1-PDF-493946<br>[58] Autonomous community of Madrid: in Primary Education (12-13 years old), the subject TECNOLOGÍA Y RECURSOS DIGITALES PARA LA MEJORA DEL APRENDIZAJE. See pages 89-90. In the first, second and third years of Compulsory Secondary Education (12-15 years old), the subject TECNOLOGÍA, PROGRAMACIÓN Y ROBÓTICA. See pages 288-295. http://www.madrid.org/ICMdownload/BXHQEMJ.pdf; https://www.bocm.es/boletin/CM_Orden_BOCM/2015/05/20/BOCM-20150520-1.PDF<br>[59] Autonomous community of Comunidad Valenciana: in the first, second and third years of Compulsory Secondary Education (12-15 years old), the subject INFORMÁTICA, pages 9-15. http://www.docv.gva.es/datos/2008/06/12/pdf/2008_7244.pdf |
| **Sweden** | - Survey of MOE<br>- Desk research | [60] Heintz, F., Mannila, L., Nygårds, K., Parnes, P., & Regnell, B. (2015). Computing at School in Sweden – Experiences from Introducing Computer Science within Existing Subjects. In A. Brodnik & J. Vahrenhold (Eds.), Informatics in Schools. Curricula, Competences, and Competitions (Vol. 9378, pp. 118–130). Cham: Springer International |

| Country | Type of information sources | Selected references from the literature |
|---|---|---|
| | | Publishing. |
| **Switzerland*** | - Survey of MOE<br>- Desk research | [61] DeutschSchweizer Erziehungsdirektoren-Konferenz. (2015). Schlussbericht der Arbeitsgruppe zu Medien und Informatik im Lehrplan 21: http://www.d-edk.ch/sites/default/files/Schlussbericht_MI_2015-02-23%20mit%20Anhang.pdf<br>[62] Broschüre der Hasler-Stiftung zu Computational Thinking in der Lehrerbildung: http://www.haslerstiftung.ch/files/webcontent/documents/d/fit_schriftenreihe/haslerstiftung_schriften04_de_v02.pdf |
| **The Netherlands** | - Desk research<br>- Interview | [63] Grgurina, N., Barendsen, E., Zwaneveld, B., van de Grift, W., & Stoker, I. (2013). Computational Thinking Skills in Dutch Secondary Education. In Proceedings of the 8th Workshop in Primary and Secondary Computing Education (pp. 31–32). New York, NY, USA: ACM. http://doi.org/10.1145/2532748.2532768<br>[64] New Tech Kids (2015). Computational thinking in the Netherlands. Retrieved from http://newtechkids.com/tag/computational-thinking/ |
| **Wales (UK)** | - Desk research | [65] Arthur, S., Crick, T., & Hayward, J. (2013). The ICT steering group's report to the Welsh government. Retrieved from http://learning.gov.wales/docs/learningwales/publications/131003-ict-steering-group-report-en.pdf<br>[66] Welsh government. (2016, July 18). Curriculum reform. Retrieved from http://gov.wales/topics/educationandskills/schoolshome/curriculum-for-wales-curriculum-for-life/?lang=en<br>[67] Welsh government. (2016, Sept 1). Digital Competence Framework. Retrieved from http://learning.gov.wales/resources/browse-all/digital-competence-framework/?skip=1&lang=en |
| **Turkey** | - Survey of MOE<br>- Desk research | [68] Gülbahar, Y., İlkhan, M., Kilis, S. & Arslan, O. (2013). Informatics Education in Turkey: National ICT Curriculum and Teacher Training at Elementary Level. In Diethelm, I., Arndt, J., Dünnebier, M. & Syrbe, J. (Eds.). Informatics in Schools: Local Proceedings of the 6th Conference ISSEP 2013 – Selected Papers (77-87). Germany: Universitätsverlag Potsdam. URL: http://pub.ub.uni-potsdam.de/volltexte/2013/6368/ |
| **Australia** | - Desk research | [69] Acara. (2015). Australian Curriculum. Digital Technologies Foundation to Year 10 scope and sequence. Retrieved from http://v7-5.australiancurriculum.edu.au/technologies/design-and-technologies/curriculum/f-10?layout=1<br>[70] Australian Council for Computers in Education. (2014). Australian Curriculum: Digital Technologies. [Video file]. Retrieved from https://www.youtube.com/watch?v=CMUkJ3qLoVw http://www.australiancurriculum.edu.au/technologies/digital-technologies/curriculum/f-10?layout=1 |
| **British Columbia** | - Desk research | [71] Ministry of Education. British Columbia (BC) (2016). Building student success. BC's new curriculum. Retrieved from https://curriculum.gov.bc.ca/curriculum/adst/6 |
| **New Zealand** | - Desk research | [72] Ministry of Education New Zealand. (n.d.). Digital Technologies Fund Opens. Retrieved August 10, 2016, from http://www.education.govt.nz/news/digital-technologies-fund-opens/ |
| **Singapore** | - Desk research | [73] Ministry of Education Singapore. (2014, September 2). Ministry of Education, Singapore: Press Releases - 42 |

| Country | Type of information sources | Selected references from the literature |
|---|---|---|
| | | Secondary Schools Offering Science, Technology, Engineering and Mathematics Applied Learning Programme (STEM ALP) [Press Release]. Retrieved from http://www.moe.gov.sg/media/press/2014/09/42-secondary-schools-offering-science-technology-engineering-and-mathematics-applied-learning-programme.php<br>[74] Cher, B. (2015). Singapore: From Smart Nation to Code Nation. Digital News Asia. Retrieved from https://www.digitalnewsasia.com/digital-economy/singapore-from-smart-nation-to-code-nation<br>[75] Kwang, K. (2016, February 27). Speaking in code: Kids in Singapore are learning a new language. Channel NewsAsia. Retrieved from http://www.channelnewsasia.com/news/singapore/speaking-in-code-kids-in/2536920.html<br>[76] Pearl, L. (2016). 19 schools to offer programming at O levels. Retrieved from http://www.straitstimes.com/singapore/education/19-schools-to-offer-programming-at-o-levels |
| **South Korea** | - Desk research | [77] Park, R. (2016). Preparing Students for South Korea's Creative Economy: The Successes and Challenges of Educational Reform. Vancouver BC, Canada: Asia Pacific Foundation of Canada. |
| **Japan** | - Desk research | [78] Plan to make programming mandatory at schools a step to foster creativity. (2016, May 18). The Japan News. Retrieved from http://the-japan-news.com/news/article/0002951918 |
| **USA** | - Desk research<br>- Interviews | [79] K12CS (2016). A Framework for K-12 Computer Science Education. Retrieved from https://k12cs.org<br>[80] Computer Science For All. (2016, January 30). [Blog post]. Retrieved from https://www.whitehouse.gov/blog/2016/01/30/computer-science-all |

LF-NA-28295-EN-N

## JRC Mission

As the science and knowledge service of the European Commission, the Joint Research Centre's mission is to support EU policies with independent evidence throughout the whole policy cycle.

**EU Science Hub**
ec.europa.eu/jrc

@EU_ScienceHub

EU Science Hub - Joint Research Centre

Joint Research Centre

EU Science Hub

Publications Office

doi:10.2791/792158

ISBN 978-92-79-64442-9