Investigation of curriculum policy on coding in six jurisdictions

NCCA
An Chomhairle Náisiúnta Curaclaim agus Measúnachta
National Council for Curriculum and Assessment

March 2018

Contents

Introduction

The purpose of this curriculum investigation is to inform the NCCA's development of guidelines to support schools' work as they pilot approaches to integrating coding into the primary school curriculum in Ireland. This piloting is the focus of phase 2 of the work with schools as part of the *Coding in Primary Schools Initiative*. To support the development of the guidelines and understand what good practice in implementing a coding or technology curriculum might look like, the NCCA has selected six jurisdictions for in-depth examination: England, Finland, New Zealand, Scotland and Northern Ireland. In the case of the sixth jurisdiction, the USA, there is no mandatory state-wide coding curriculum. Instead, it was necessary to look at recommendations made by the Computer Science Teachers Association (CSTA). These are looked at in the context of how they are being used and implemented in Washington State's primary curriculum.

New Zealand, Scotland, England and the USA were part of the NCCA's 2016 desktop audit of coding in the primary curriculum in 22 jurisdictions. These four jurisdictions with the addition of Finland and Northern Ireland have been selected for more in-depth analysis because of their uniqueness with regard to the early introduction of coding as a component of a mandatory separate computing curriculum or an integrated cross-curriculum programme at primary level. They also, collectively, represent a range of different pedagogical approaches to implementing coding in their curricula. Their strong emphasis on research-driven approaches, and their reputation as leading trends in technology and ICT integration within primary curriculum policy, is also evident.

With the purpose of informing the development of guidelines for phase 2 of the school-based initiative, some of the key questions guiding the curriculum investigation include:

- Can common coding standards, strands or themes be identified that run through the curriculum of each of the six selected jurisdictions?
- At what age/level is coding integrated in the primary curricula and how is coding defined?
- What is the scope of children's learning?
- Does teaching coding have the potential for cross-curriculum opportunities?
- How are schools, with limited resources for technology, supported in implementing the curriculum?
- How is continuing professional development provided for teachers? What does it entail?

When looking more closely at coding in the six jurisdictions, it becomes clear from the outset that 'coding' can be part of a much broader subgroup of titles such as 'Computer Science' (CS), 'Computational Thinking' (CT) and 'Information and Communications Technology' (ICT). For the purposes of this investigation, coding is the focus of the report but references to the subgroups is unavoidable.

Code in the curriculum

In many countries around the world there is a growing recognition that Computer Science has both enormous educational benefits such as thinking and problem-solving skills, understanding a world suffused with digital technology, as well as economic benefits as companies struggle to recruit well-educated graduates. The introduction of Computer Science, Programming/Coding, and Computational Thinking to primary school curricula has been achieved in several countries and is being considered in many more. As this is a relatively recent trend in curriculum development there is still much to learn about the purpose of this area of learning in a child's primary education, how best to design and integrate this area into a primary curriculum, and how best to use these curricula in schools.

The motivation for teaching topics such as coding, specifically at primary school level, is largely based on the value attached to exposing children to the concepts, and developing awareness, of the technology which surrounds them in the world in which they live; for example, the Finnish approach to coding sees it as a modern method of creativity and self-expression that also happens to prepare children for the modern technical society (The Finnish National Board of Education).

In addition to general thinking and collaboration skills, benefits of a Coding and Computational Thinking-based curriculum for this age group are:

- Diversity: the importance of reaching children aged 12 or younger, particularly females, when they are forming views of their competence and general ability in this subject
- Ability to learn: children learn languages better at a young age, and this may transfer to learning the language of computing
- Empowering children to change the world by giving them a sense that they can be more than just users of digital devices
- Preparing children for future endeavours in computing.

  (In Bell (ed) (2015, pp.2-3, Margolis and Fisher; Johnson and Newport; Lee; Martin and Apone; Phillips).

Curriculum comparison

Making comparisons of international curricula is not straightforward because of the diversity of education systems and the socio-cultural and political contexts in which they are embedded. The development of computing or technology as an area in primary school curricula is a fairly recent trend internationally and has drawn on the work of established experts in the field. As a result, significant similarities can be found in a number of international curricula including the six spotlighted in this audit.

However, ‘coding’ is not mentioned in isolation in any of the six curricula investigated. Instead, it is associated with ‘Computer Science’, ‘Computational Thinking’ and ‘Information and Communications Technology'. Despite this, coding can be seen as the primary focus when the similarities listed below are taken into consideration.

In the countries investigated, creating with technology, understanding technology and using technology are all named very differently, but are all linked when children learn how to code. For the purposes of this audit, examples of creating, understanding and using technology are presented under three headings—Coding/Computational Thinking, Information and Communications Technology, and Human Interaction with Technology—as seen in the tables associated with each jurisdiction.

England

The English National Primary Curriculum provides pupils with an introduction to the essential knowledge that they need to be educated citizens. Every state-funded school must offer a curriculum which is balanced and broadly-based and which:

- promotes the spiritual, moral, cultural, mental and physical development of pupils at the school and of society, and
- prepares pupils at the school for the opportunities, responsibilities and experiences of later life.

Maintained schools in England are legally required to follow the statutory national curriculum which sets out in programmes of study, on the basis of key stages, subject content for those subjects that should be taught to all pupils. All schools must publish their school curriculum by subject and academic year online.[1] There are twelve areas of learning within the English primary curriculum:

- English
- Maths
- Science
- Art and design
- Citizenship
- Computing
- Design and technology
- Geography
- History
- Languages
- Music
- Physical Education (PE)

Computing Curriculum

The broad subject of computing—incorporating the three areas of computer science, digital literacy and information technology (IT)—has become mandatory in schools for children aged five to 16 years. The current Computing Curriculum covers all three strands and is non-prescriptive in terms of coding languages, software or hardware.

By the end of each key stage (1 and 2 for primary level), pupils are expected to know, apply and understand the matters, skills and processes specified in the curriculum's Programme of Study which can be seen in Tables 1 and 2. The Computing Curriculum aims to ensure that all pupils:

- Can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
- Can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
- Can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
- Are responsible, competent, confident and creative users of information and communication technology.

One major difference between the Computing Curriculum introduced in 2014 and the previous ICT Curriculum is the new emphasis on the principles and concepts of computer science, alongside digital literacy and IT. This means children gain the skills needed to represent real-world problems in a form applicable to computational investigation together with the skills needed to explore those

[1] Source: Department of Education (England) https://www.gov.uk/government/publications/national-curriculum-in-england-primary-curriculum

representations to develop algorithmic solutions, and practical experience of writing computer programs implementing those solutions. These could include creating simple algorithmic examples or a sequence of instructions for everyday activities such as the steps involved in how to make toast.

Computational thinking is a key theme of the Computing Curriculum in England, appearing in the first sentence of the purpose of study (Department of Education, England 2013, p.1): *A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world*. There is a hope in England that starting computing education in primary school has the potential to encourage more children to take it to a higher level and reduce the proportion of children that perceive computing as 'not for them'. In addition, when computing is embedded within other subjects in primary school, there is potential to give children an understanding of the subject's relevance and its potential applications.

Table 1 – Key Skills. England: Foundation (ages 3-5), Key stage 1 (ages 5 to 7)

Coding/Computational Thinking	Information and Communications Technology	Human Interaction with Technology
▪ Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions ▪ Create and debug simple programs ▪ Use logical reasoning to predict the behaviour of simple programs	▪ Create, organise, store, manipulate and retrieve content ▪ Use technology purposefully to create, organise, store, manipulate and retrieve digital content	▪ Recognise common uses of information technology beyond school ▪ Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies

Table 2 – Key Skills. England: Key Stage 2 (ages 7 to 11)

Coding/Computational Thinking	Information and Communications Technology	Human Interaction with Technology
▪ Decompose problems; explain and correct errors in algorithms	▪ Select, use and combine software to work with data; use search technology; create content	▪ Use safely, respectfully and responsibly; appropriate

▪ Use sequence, selection and repetition in programs; work with variables and various forms of input and output ▪ Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts ▪ Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs	▪ Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information ▪ Understand computer networks including the internet; how they can provide multiple services, such as the World Wide Web	behaviour; reporting concerns ▪ Be discerning in evaluating digital content ▪ Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact

New Zealand

The New Zealand Curriculum identifies five key competencies; these are not separate or stand-alone but instead, are key in every learning area. These competencies continue to develop over time shaped by interactions with people, places, ideas, and things. The competencies are

- Thinking
- Using language, symbols and texts
- Managing self
- Relating to others
- Participating and contributing.[2]

There are eight areas of learning within the New Zealand Curriculum:

- English
- The Arts
- Health and Physical Education
- Learning Languages
- Mathematics and statistics
- Science
- Social Sciences
- Technology.

Digital Technology Curriculum

New Zealand revised their technology learning area and introduced the Digital Technologies Curriculum to enable their students to become innovative creators of digital solutions. The newly introduced (2018) digital technology learning area was revised from the previous curriculum to enable students to build their skills, so they can become innovative creators of digital solutions – moving beyond solely being users and consumers of digital technologies.[3] The revised technology learning area has three strands and five technology areas:

- Technological Practice
- Technological Knowledge
- Nature of Technology.

The three strands are embedded within each of the five technological areas:

[2] Source: Ministry of Education (New Zealand) http://nzcurriculum.tki.org.nz/
[3] http://nzcurriculum.tki.org.nz/The-New-Zealand-Curriculum/Technology

- Computational thinking for digital technologies
- Designing and developing digital outcomes
- Designing and developing materials outcomes
- Designing and developing processed outcomes
- Design and visual communication.

In Technological Practice, students examine the practice of others and undertake their own. They develop a range of outcomes including concepts, plans, briefs, technological models, and fully realised products or systems. When learning about Technological Knowledge, students build an understanding in relation to how and why things work. They come to understand how constituent parts of systems work together and how this is essential to understanding how and why systems operate in the way they do. For the Nature of Technology, students learn to critique the impact of technology on societies and the environment and to explore how developments and outcomes are valued by different peoples in different times.

Computational thinking for digital technologies is named as one of five technological areas in the primary curriculum and is where students develop algorithmic thinking skills and an understanding of computer science principles that underpin all digital technologies. Students also learn core programming concepts so that they can become creators of digital technologies, not just users.

The new Technology Curriculum has been in development since 2014 and was completed at the end of 2017. During 2018 and 2019, programmes of support and teacher CPD initiatives will help teachers and schools prepare for working with the new curriculum content. The curriculum will be become statutory in all schools by 2020. The change signals the need for greater focus on students building their skills so that they can be innovative creators of digital solutions, moving beyond solely being users and consumers of digital technologies.

The key underlying principles of the new Technology Curriculum are to develop digitally capable thinkers, producers and creators and to show students how digital technologies work and how they (students) can use that knowledge to solve problems so they can become creative innovators of digital solutions. In the case of primary schools, teachers are advised to generally take a cross-curriculum approach with students learning in the three technology strands as part of a topic or theme that encompasses several curriculum learning areas. Students work towards ‘progress outcomes’ which can be seen in Tables 3 and 4. These outcomes describe the significant learning steps that students take as they develop their expertise in computational thinking.

Table 3 – Key Skills. New Zealand: Level 1, Years 1-3 (ages 5 to 7)

Coding/Computational Thinking	Information and Communications Technology	Human Interaction with Technology
▪ Students use decomposition skills to break down simple non-computerised tasks into precise, unambiguous, step-by-step instructions (algorithmic thinking) ▪ Students give these instructions, identify any errors in them as they are followed, and correct them (simple debugging) ▪ Students give, follow and debug simple algorithms in computerised and non-computerised contexts ▪ Students uses algorithms to create simple programs involving outputs and sequencing (putting instructions one after the other) in age appropriate programming environments	▪ Use common input/output devices (e.g. keyboard, pointing device, touch screen) ▪ Create, organise, store, manipulate and retrieve content ▪ Describe hardware and software components; input/output devices ▪ Create, organise, store, manipulate and retrieve content, including multi-media	▪ Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies; recognise use beyond school ▪ Ergonomics, digital devices in everyday life; ethical and safe use

Table 4 – Key Skills. New Zealand: Levels 2, 3, year 3-8 (ages 7 to 12)

Coding/Computational Thinking	Information and Communications Technology	Human Interaction with Technology
▪ Students decompose problems into step-by-step instructions to create algorithms for computer programs ▪ They use logical thinking to predict the behaviour of the programs, and they	▪ Peripheral devices; data capture; data transfer ▪ Select, use and combine software to work with data; use search technology; create content, simple spreadsheets and charts ▪ Identify simple hardware and software	▪ Use safely, respectfully and responsibly; appropriate behaviour; reporting concerns ▪ Understand how computer systems meet community and personal needs;

understand that there can be more than one algorithm for the same problem ▪ Students develop and debug simple programs that use inputs, outputs, sequence and iteration (repeating part of the algorithm with a loop) ▪ Students decompose problems to create simple algorithms using the three building blocks of programing: sequence, selection, and iteration ▪ Students debug simple algorithms and programs by identifying when things go wrong with their instructions and correcting them, and they are able to explain why things went wrong and how they fixed them ▪ They understand how computers store more complex types of data using binary digits, and they develop programs considering human-computer interaction (HCI) heuristics	problems; understand networks ▪ Create, organise, manipulate and retrieve content	evaluate adequacy of a solution; collaboratively plan creation and communication of information; ethical decisions and behaviour; social media ▪ Studenst demonstrate safe/ethical use of digital devices ▪ Understand the impact of computers on humans

Finland

In Finland the National Core Curriculum for Basic Education in primary schools is underpinned by transversal competences which consist of knowledge, skills, attitudes and will. Competence also means an ability to apply knowledge and skills in a given situation. Each subject in the curriculum builds pupils' competence through the contents and methods typical of its field of knowledge.

There are seven transversal competencies:

- Thinking and learning to learn
- Cultural competence, interaction and self-expression
- Taking care of oneself and managing daily life
- Multi-literacy
- ICT Competence
- Working life competence and entrepreneurship
- Participation, involvement and building a sustainable future.

There are twelve areas of learning within primary schools' Finnish National Core Curriculum for Basic Education:

- Mother Tongue and Literature
- Second National Language
- Foreign Languages
- Mathematics
- Environmental Studies
- Religion
- Ethics
- Visual arts
- Crafts
- Physical Education
- Guidance Counselling

History and Social Studies is an additional curriculum area as students move into grades 3-6 (ages 7-11).[4]

[4] Source: National Board of Education (Finland) http://www.oph.fi/english/curricula_and_qualifications/basic_education

Embedded Technology across the curriculum

> *Opportunities for pupils to develop their information and communication technology skills have been improved in all subjects, with technology being included more in instruction and study. --- Technology plays an increasingly significant role in everyday school routines, thus allowing pupils to be more easily involved in the development and selection of their own learning environments*. (Source: National Board of Education (Finland) http://www.oph.fi/english/curricula_and_qualifications/basic_education)

In Finland, the decennial reform process of the National Core Curriculum for Basic Education (primary and lower secondary) took place during 2012–2014. The new local municipality and school curricula, crafted on the basis of the national guidelines, were implemented from August 2016. The new national curriculum emphasizes, alongside subject studies, developing transversal competences including ICT Competence as noted earlier. This competence is divided to four main areas of development:

- Students understand the principles and basic concepts of using ICT, and learn to develop practical ICT-skills via producing own content.
- Students are encouraged to use ICT responsibly, safely and ergonomically.
- Students are taught to use ICT in database management and in creative efforts.
- Students gain experiences and practice using ICT in communication and networking.

The new National Core Curriculum for Basic Education framework for primary education in Finland (from 2016 onwards) states that programming (or coding) is part of all education. Information technology is not seen as a separate entity, it is interwoven within all subjects as one of the seven transversal competencies. Coding and computational thinking are mainly found under the terms 'programming' and 'algorithmic thinking'. Programming has been integrated into the new National Core Curriculum for Basic Education framework as part of the general ICT competence as well as part of mathematics beginning in the first grade, and of crafts starting from the third grade.

There are specific goals for the ICT competence for grades 1–2 and for grades 3–6 which can be seen in Tables 5 and 6. According to these goals, pupils start their journey in the world of programming by learning age-suitable programming, starting from the 3rd grade by focusing how human decisions influence the outcomes in programming, and eventually developing their coding skills as part of multiple school subjects.

As for programming in Mathematics, the emphasis is on developing algorithmic as well as computational thinking skills. Pupils learn to first create basic sequences of instructions, then progress to learning how to code their own programs in a visual programming environment, and finally, to applying the principles of algorithmic thinking into programming simple programs.

Table 5 – Key Skills. Finland: (ages 5 to 7)

Coding/Computational Thinking	Information and Communications Technology	Human Interaction with Technology
▪ Pupils prepare step-by-action instructions and act according to the instructions ▪ Pupils know that programming begins with the elaboration of step-through procedures that are also tested ▪ Pupils receive and share experiences of working with digital media as well as programming appropriate for their age ▪ Students get to know the basics of programming by forming and testing sequential instructions	▪ Pupils practice using devices, software, and services and learn their key use and operating principles ▪ Students practice keyboard skills and other basic text production and processing skills ▪ Pupils gain and share experiences of working with digital media ▪ Gamification is used to promote learning	▪ Together with the teacher, pupils search for safe ways of using ICT and the related etiquette ▪ Attention is paid to a good working posture and the appropriate duration of working periods and their effects on student well-being ▪ Pupils gain experience in using social networking services that support learning as well as practise using ICT in different interactive situations

Table 6 – Key Skills. Finland: (ages 7 to 12)

Coding/Computational Thinking	Information and Communications Technology	Human Interaction with Technology
▪ Students draw up operational instructions as computer programs in a graphical programming environment ▪ Pupil knows how to program an operating program in a visual	▪ Pupils practice producing and processing texts fluently using different tools and learn about working with image, sound, video, and animation ▪ The pupils learn to use different devices, software, and services	▪ Pupils examine the impacts of ICT on their daily lives and find ways of using it sustainably ▪ Pupils are guided in responsible and safe use of ICT, good manners, and knowing basic copyright principles

programming environment ▪ When testing programming, students learn how technology depends on human solutions ▪ Students practice physical programming, with automation and robotics ▪ Students plan and execute programs in a visual programming environment	and to understand the logic of their use and operation ▪ Pupils are encouraged to use ICT to implement their ideas independently and together with others	▪ Pupils practise using various communication systems and educational social media services ▪ Pupils obtain knowledge and experience of the health impacts of good working postures and working periods of a suitable length ▪ Pupils are guided to act according to their roles and the characteristics of the device they are using as well as to take responsibility for their communication

CSTA Standards Computer Science Teachers Association Grade K-2

USA - Washington State[5]

There is currently no national curriculum in the United States. States, school districts and national associations do require or recommend that certain standards be used to guide school instruction. This section of the report looks at recommendations by the Computer Science Teachers Association (CSTA) which is a professional association that supports and promotes the teaching of computer science.

Washington State is used below as a case study to explore how the CSTA recommended standards have been applied to the State's Computer Science Curriculum for primary school children.

[5] The information presented is based on that drawn from *Superintendent of Public Instruction – Washington State.*

Computing Science Curriculum

In December 2016, Washington State required the adoption of nationally-recognised computer science standards and are currently encouraging schools to adopt the curriculum guidelines and to integrate them into their classrooms. The State is committed to implementing high-quality computer science instruction to:

- Increase the opportunity for all students to gain knowledge of computer science.
- Introduce the fundamental concepts and applications of computer science to all students, beginning at the elementary school level.
- Make computer science at the secondary level accessible, worthy of a computer science credit, and/or equivalent to math and science courses as a required graduation credit (see Level 3B of computer science standards).
- Offer additional secondary-level computer science instruction that allows interested students to study facets of computer science in depth and prepare them for entry into a career or college.

The computer science concepts, standards and practices developed are intended to empower students to:

- Be informed citizens who can critically engage in public discussion on computer science related topics
- Develop as learners, users, and creators of computer science knowledge and artefacts
- Better understand the role of computing in the world around them
- Learn, perform, and express themselves in other subjects and interests
- Increase career and college readiness.

All curriculum in Washington State is decided locally, within each district. Districts can decide how to incorporate the Computer Science K-12 learning standards into each grade level; these levels can be seen in Tables 7 and 8. The learning standards are meant to establish a baseline literacy in computer science and consist of five core concepts and seven core practices, as listed below.

Core Concepts

- Computing Systems
- Network and the Internet
- Data and Analysis
- Algorithms and Programming
- Impacts of Computing

Core Practices

- Fostering an Inclusive and Diverse Computing Culture
- Collaborating
- Recognizing and Defining Computational Problems
- Developing and Using Abstraction
- Creating Computational Artefacts
- Testing and Refining
- Communicating.

Currently one in ten schools offers programs that meet the standards and the state hopes that over half of the schools will be meeting the standards by 2019. The Washington-based standards introduce fundamental computer science concepts from the beginning of primary school to all students and the schools are hoping to present computer science in way that can fulfil a computer science, math or science credit for graduation.

The State is particularly mindful of the need to encourage participation of groups that are underrepresented in the technology field, including females. The following is noted under the title 'Learning standards: Equity, Access, Inclusion, and Diversity' in their computer standards document:

> *Computer science, among other STEM disciplines, can provide the knowledge and skills to empower individuals to create technologies with broad influence and impact. Women, underrepresented minorities, and people with disabilities are often missing in computer science classes, majors, and occupations*.

Computational thinking is seen as an overarching practice reflected in a number of the core computer science practices. The standards exist to provide a framework for practitioners to reference as they design curriculum and testing around computer science. As with all of Washington's learning standards, they are written in a way that allows local-decision making. School districts are guided to make decisions and design curriculum that are appropriate for their districts, communities, and classrooms. The standards help define what is important for students to know and be able to do as they progress through school.

Table 7 – Key Skills. CSTA – Computer Science Teachers Association Grade K-2 (ages 5 to 8)

Coding/Computational Thinking	Information and Communications Technology	Human Interaction with Technology
▪ Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks ▪ Develop programs with sequences and simple loops, to express ideas or address a problem ▪ Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions ▪ Develop plans that describe a program's sequence of events, goals, and expected outcomes ▪ Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops ▪ Using correct terminology, describe steps taken and choices made during the iterative process of program development	▪ Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use. ▪ Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware) ▪ Describe basic hardware and software problems using accurate terminology. ▪ Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data ▪ Collect and present the same data in various visual formats ▪ Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions	▪ Compare how people live and work before and after the implementation or adoption of new computing technology. ▪ Work respectfully and responsibly with others online. ▪ Keep login information private, and log off of devices appropriately

Table 8 – Key Skills. CSTA: Computer science Teachers Association: Grade 3-5 (ages 8 to 12)

Coding/Computational Thinking	Information and Communications Technology	Human Interaction with Technology
▪ Decompose (break down) problems into smaller, manageable subproblems to	▪ Describe how internal and external parts of computing devices	▪ Discuss computing technologies that have changed the world, and express

facilitate the program development process ▪ Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features ▪ Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences ▪ Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended ▪ Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development ▪ Describe choices made during program development using code comments, presentations, and demonstrations ▪ Compare and refine multiple algorithms for the same task and determine which is the most appropriate ▪ Create programs that use variables to store and modify data ▪ Create programs that include sequences, events, loops, and conditionals	function to form a system. ▪ Model how computer hardware and software work together as a system to accomplish tasks ▪ Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.	how those technologies influence, and are influenced by, cultural practices. ▪ Use public domain or creative commons media, and refrain from copying or using material created by others without permission. ▪ Discuss real-world cybersecurity problems and how personal information can be protected ▪ Keep login information private, and log off of devices appropriately

Northern Ireland

The Northern Ireland Primary Curriculum aims to empower young people to achieve their potential and to make informed and responsible decisions throughout their lives.[6] In Foundation Stage (ages 4-6), Key Stage 1 (ages 6–8) and Key Stage 2 (ages 8–11) the curriculum is broad and balanced. Every child has the opportunity to build on the skills they have acquired and developed in Foundation Stage, so they can experience success in learning and achieve as high a standard as possible.[7]

There are six areas of learning within the Northern Ireland Curriculum including:

- Mathematics and Numeracy
- Language and Literacy
- The Arts (including Art and Design, Drama and Music)
- The World Around Us (focusing on the development of knowledge, skills and understanding in Geography, History and Science and Technology)
- Personal Development and Mutual Understanding (PDMU)
- Physical Education (PE).

Embedding Technology across the curriculum

There is no specific technology or computing curriculum in the Northern Irish Primary Curriculum. ‘Using ICT’ is embedded across the curriculum. In this way, the skills of ‘Using ICT’ are seen as integrated within classroom work and are not taught in isolation. The Northern Ireland Curriculum supports the development of skills and capabilities including Cross-Curricular Skills (CCS) and Thinking Skills and Personal Skills (TSPC) in ‘Communication’, ‘Using Mathematics’ and ‘Using ICT’. These are divided into five strands:

- Managing Information
- Thinking, Problem-Solving, Decision-Making
- Being Creative
- Working with Others
- Self-Management.

[6] Source: Department of Education (Northern Ireland) http://ccea.org.uk/sites/default/files/docs/curriculum/area_of_learning/ks1_2_bigpicture.pdf

[7] Source: Department of Education (Northern Ireland) http://ccea.org.uk/curriculum/key_stage_1_2/overview

Children in primary schools and students in post-primary schools are expected to develop the skills of 'Using ICT' by taking part in meaningful research and purposeful activities set in relevant contexts. The curriculum states that they should use ICT to handle and communicate information, solve problems, pose questions and take risks. They should process, present and exchange their ideas and translate their thinking into creative outcomes that show an awareness of the audience and purpose. They should also use ICT to collaborate within and beyond the classroom, to share and exchange their work. In developing their ICT skills through these experiences, children/students focus on the five areas of Explore; Express; Exchange; Evaluate; and Exhibit.

It is within the 'Using ICT' curriculum specification and guidance that the elements of programming and computing are found. The guidance for 'Using ICT' gives levels of progression (levels 1 to 5 for primary school) which can be seen in Tables 9 and 10. There are also non-statutory 'Desirable Features' which have been subdivided into the eight sections; Desktop Publishing, Film and Animation, Interactive Design, Managing Data, Music and Sound, On-line Communication, Presenting and Working with Images. The programming and computing elements are primarily within 'Interactive Design'.

While the phrase 'Computational Thinking' does not explicitly appear within the Northern Ireland Curriculum, the accepted elements are comfortably within the 'Thinking Skills and Personal Capabilities' section of the curriculum, in particular, 'Thinking, Problem Solving and Decision Making' and 'Being Creative'. As in 'Using ICT', these are also addressed in a cross-curricular manner and can be found within different curriculum areas. The curriculum requirements for using ICT are set out under headings often described as the 5 'E's: Explore, Express, Exchange, Evaluate and exhibit.

While coding is not included in the curriculum, there is flexibility for schools to choose to teach it. It has been suggested that if Northern Ireland introduce a mandatory computing curriculum it should ensure that it incorporates a balance of computer science, digital literacy and information technology. The last Minister for Education stated that while he did not plan to reference coding within the curriculum, there could be scope for consideration of its inclusion when the curriculum is next subject to review.

Table 9 – Key Skills. Northern Ireland: Foundation Stage (ages 4-6) and Key Stage 1 (ages 6 to 8)

Coding/Computational Thinking	Information and Communications Technology	Human Interaction with Technology
▪ Explore a digital device or environment using simple commands ▪ Enter commands to create movement or change ▪ Talk about what they have done ▪ Solve problems using a digital device or environment ▪ Input sequences of commands; and make modifications to improve their work	▪ Find and select information from a given digital source ▪ Express ideas by creating pictures and composing text or adding their own voiceover	▪ Know that digital methods can be used to communicate ▪ Identify and talk about ways of communicating digitally

Table 10 – Key Skills. Northern Ireland: Key Stage 2 (ages 8 to 11)

Coding/Computational Thinking	Information and Communications Technology	Human Interaction with Technology
▪ Investigate and solve problems in a digital environment ▪ Input more complex sequences of commands ▪ Use appropriate ICT tools and features to improve work ▪ Plan and solve a more complex problem or design and create an interactive 'product' demonstrating a clear sense of purpose and audience ▪ Carry out ongoing improvements and evaluate their process and outcome	▪ Research, select, edit and use information from given digital sources ▪ Carry out and edit a series of instructions, make predictions and solve problems using a digital device or environment ▪ select, organise, store and retrieve their work to showcase learning digitally in a personalised area	▪ Pupils should demonstrate, when and where appropriate, knowledge and understanding of e-safety including acceptable online behaviour ▪ Pupils should be provided with opportunities to develop knowledge and understanding of e-safety and acceptable online behaviour

Scotland

Curriculum for Excellence, the Scottish curriculum for 3-18 year-olds, aims to help children and young people gain the knowledge, skills and attributes needed for life in the 21st century including skills for learning, life and work. In doing this, the curriculum enables children and young people to become:

- Successful learners
- Confident individuals
- Responsible citizens
- Effective contributors.

These are referred to as the four capacities.[8]

There are eight areas of learning within *Curriculum for Excellence*:

- Expressive arts
- Health and wellbeing
- Languages
- Mathematics
- Religious and moral education
- Sciences
- Social studies
- Technologies.

Each area has a set of 'Experiences' and 'Outcomes' (Es and Os).

Computing Science Curriculum

Computer Science sits within the Technologies section of the *Curriculum for Excellence*. The technologies framework provides a range of different contexts for learning including the themes across learning; learning for sustainability, global citizenship, enterprise, that draw on important aspects of everyday life and work.[9] There are three Significant Aspects of Learning or 'SALs' for the Computing Science Curriculum:

- Understanding the world through computational thinking
- Understanding and analysing computing technology
- Designing, building and testing computing solutions.

[8] Source: Department of Education (Scotland) http://www.gov.scot/Topics/Education/Schools/curriculum
[9] Source: https://education.gov.scot/Documents/Technologies-es-os.pdf

Recently, the 'Significant Aspects of Learning' (SALs) specifically for the Computer Science Curriculum for primary schools have been introduced as overarching themes across groups of Experiences and Outcomes, (Es and Os). Benchmarks for key stages 1 and 2 can be seen in Tables 11 and 12. Technology benchmarks for the Computing Science Curriculum set out clear statements about what learners need to know and be able to do. These are intended to be a practical resource to help teachers make judgements about children's learning and how this can be progressed.

An innovative contribution of the Computer Science Curriculum (compared to curricular frameworks worldwide) is the organisation of core computational thinking (CT) concepts according to three SALs. The first SAL introduces learners to core concepts in Computational Thinking and Computer Science. The second SAL introduces learners to how tools and languages use those concepts. The third SAL sees learners apply their learning of the concepts of Computational Thinking and the basics of Computer Science from the first two SALs in order to be creative with technology or solve problems.

- **SAL 1, Theory**: Understanding the world through computational thinking and knowledge of core computing science concepts is necessary in order to later apply that knowledge using languages and technology.
- **SAL 2, Languages and Tools**: Understanding of computing technology and the programming languages that control them is essential before designing and building using these tools.
- **SAL 3, Creating**: Using conceptual and technological knowledge to design, build and test.

The SALs for Computing Science build upon each other. The conceptual knowledge gained when working towards the first SAL, 'understanding the world through computational thinking', is required to then understand computing languages and technologies in the second SAL, before 'design, build and test computing solutions' in the final SAL using those technologies.

Table 11 – Key Skills. Scotland: Key Stage 1, Early, First and Second levels (ages 4 to 7)

Coding/Computational Thinking	**Information and Communications Technology**	**Human Interaction with Technology**
▪ Identifies and sequences the main steps in an everyday task to create instructions / an algorithm ▪ Classifies objects and groups them into simple categories	▪ Computers take in inputs, process them, store information, and then output the results ▪ Demonstrates an understanding of how symbols can represent process and information	▪ Identifies computing devices in the world (including those hidden in appliances and objects such as automatic doors) ▪ Can explore, play and communicate using digital technologies safely and securely

▪ Identifies patterns, similarities and differences in objects or information ▪ Designs a simple sequence of instructions/algorithm for programmable device to carry out a task ▪ Identifies and corrects errors in a set of instruction ▪ Creates programs in a visual programming language including variables and conditional repetition ▪ Explains and predicts what a program in a visual programming language will do when it runs ▪ Explains and predicts how parallel activities interact	▪ Describes the purpose of the processor, memory and storage and the relationship between them ▪ Can use digital technologies to explore how to search and find information ▪ Understand how information is stored and how key components of computing technology connect and interact through networks	▪ Can use digital technologies to search, access and retrieve information being aware that not all of this information will be credible

Table 12 – Key skills. Scotland: Key Stage 2 (ages 7 to 11)

Coding/Computational Thinking	Information and Communications Technology	Human Interaction with Technology
▪ Identifies the use of common algorithms such as sorting and searching as part of larger processes ▪ Can informally compare algorithms for correctness and efficiency ▪ Designs and builds a program using a visual language combining constructs and using multiple variables ▪ Creates a design using accepted design methodologies for example, pseudocode, storyboarding, structure diagram, data flow diagram, flow charts ▪ Debugs code and can distinguish between the nature of identified errors e.g. syntax and logic ▪ Writes code in which there is communication between parallel processes (in a visual language) ▪ Writes code which receives and responds to real world inputs (in a visual language) ▪ Writes code which receives and responds to real world inputs (in a visual language)	▪ Can describe the structure and operation of computing systems which have multiple software and hardware levels that interact with each other ▪ Can select appropriate development tools to design, build, evaluate and refine computing solutions based on requirements ▪ Can use digital technologies to process and manage information responsibly and can reference sources accordingly ▪ Demonstrates an understanding of the basic principles of compression and encryption of information ▪ Can select and use digital technologies to access, select relevant information and solve real world problems	▪ Having used digital technologies to search, access and retrieve information can justify selection in terms of validity, reliability and have an awareness of plagiarism ▪ Can explore the impact of cyber-crime for business and industry and the consequences this can have on others ▪ Can explore online communities demonstrating an understanding of responsible digital behaviour and is aware of how to keep themselves safe and secure

Comparing the curricula in the six jurisdictions

As noted previously, 'coding' is not mentioned in isolation in any of the six curricula investigated. Instead, it is associated with 'Computer Science', 'Computational Thinking' and 'Information and Communication Technology'. Despite this, coding can be seen as the primary focus when the common features listed below are taken into consideration. Looking at the six curricula, there are a number of similarities:

- The necessity to develop a curriculum that is fit-for-purpose that gives children the skills needed for living in the 21st century.
- Providing children, from an early age, with experiences of using, creating and understanding about how technology works.
- The ongoing review of technology/computing curricula in order to keep up with an ever-changing technological environment.
- Technology, or computer studies, to be a cross-curriculum programme of study integrated throughout many other curriculum subjects.
- Collaboration with the tech industry, in an advisory role, in the development of technology/computer science as a subject in primary school and, in some cases, supporting the continued professional development of teachers.
- Importance of supporting and developing a clear strategy for training teachers in implementing a technology curriculum and ongoing review of the types of continued professional development offered to teachers.
- The development of pedagogical approaches which include play, team work, creativity and real-world problem-solving when teaching about technology/computer science, in particular within the early years.
- When referenced within a curriculum, the inextricable linking of coding/programming with computational thinking.

What is taught in relation to coding and from what age?

All six curricula introduce computational thinking activities and coding concepts from the first year when children attend primary school. Children develop and enhance these skills with age-appropriate tasks and resources as they progress through school.

From the beginning of primary, the six jurisdictions introduce coding and computational thinking concepts using algorithms, patterns, logic, decomposition and debugging which are based on concrete physical motion that children can relate to. For example, their problem might be how to navigate a route: they think about how to solve it, press the keys (on a Beebot) or give instructions for classmates to follow and then debug as they go along. A combination of devices such as 'Beebots' and 'Scratch Jnr' are used. Across the ages 7 and 11, a continuation of more complicated uses of algorithms, debugging, logic, and the introduction of decomposition, repetition, sequences and variables occurs. In addition, children are generally introduced to a block-based programming language such as Scratch.

There are clear progressions in learning regarding coding/programming skills which can be seen in Tables 1-12. These progressions are very similar for all countries. This convergence in design of curricula is likely due to the large amount of recent international research which is readily available. In view of this recent research, five of the six jurisdictions, England being the exception, either amended or developed completely new curricula within the last two years. England's Computing Curriculum became mandatory in 2014. Research is ongoing to explore the most effective way of introducing computational thinking and coding to children in the early years of primary.

Resources

A review of the curricula illustrates a commonality in the types of teaching and learning resources used with 'Scratch Jnr' (a block-based programming language) being used in the junior classes as it does not rely on children being literate to use the software. 'Scratch' which is a more developed programming tool, tends to be used by older children as noted in the report, *Provision of Courses in Computer Science in Upper Second Level Education Internationally* (LERO, The Irish Software Research Centre, 2015) completed for the NCCA. Schools and teachers also use other freely available software in conjunction with activities such as the 'Hour of Code' and 'Code.org' to teach the curriculum. There is also growing evidence of the use of 'unplugged' activities where children don't access computers to learn the fundamentals of computer science. Unplugged activities, which is a relatively new strategy to teaching computer science internationally, teaches coding and computational thinking skills through games and puzzles that use cards, string, crayons and other materials that schools are already very familiar with and which are readily available. ‘Barefoot Computing’ funded by British Telecom, has aligned resources with the primary computer science curriculum and is used in Northern Ireland, England and Scotland. It is also freely available worldwide. ‘Hello Ruby’ created by Finnish author Linda Liukas is another example of unplugged Computer Science and has links to the Finnish curriculum and is also used in over 40 countries.

There are no prescribed or mandatory resources associated with the curriculum in any of the jurisdictions examined. There are obvious benefits to this, however, there is also growing concern expressed by teachers. Their concerns centre on the fact that, as there are more and more resources becoming available, a significant amount of their time is taken up with deciding how useful the resources are: more advice, at state level, is required.

For the teaching of coding at primary school level, there are many resources and guides available, and dozens of programming languages designed specifically for teaching this age group. There are also repositories of resources that can guide how these topics are taught at primary level, for example, the CSTA repository, Barefoot Computing and CS Unplugged online resource.

Implications for curriculum integration

England, Scotland and New Zealand, and the CSTA recommendations for Washington State all have ‘Computing’, ‘Computer Science’, ‘Technology Curriculum’ and ‘Computer Science’ listed as individual curriculum subjects. However, they all recommend that computer science cannot be understood unless integrated with other curriculum subjects and real-world scenarios.

Finland and Northern Ireland have taken a different approach as coding is not seen as a separate subject or part of a separate subject, and they also advocate that learning about technology should be integrated into other subjects. Both countries, however, include coding and computational thinking skills as part of their Maths Curriculum. In Finland, programming and algorithmic thinking are mandatory but not in Northern Ireland. In both Finland and Northern Ireland, Information and Communications Technology is integrated throughout most subjects.

Designing a primary school coding curriculum

Coding and computational thinking engages sequential and logical thinking, and this kind of thinking is also involved in literacy and maths. In all six jurisdictions, some aspects of learning to code have been directly linked to their Maths Curriculum. Finland, Northern Ireland and the countries which have computing as a standalone subject, encourage the integration of coding skills into other curriculum subjects. They note the importance of strengthening the overall curriculum while introducing coding and computational thinking to foster positive attitudes towards computer science at an early age. The CSTA K-12 Computer Science standards have five strands, or themes, that run through the curriculum: computational thinking, collaboration, computing practice and programming, computer and communication devices, and community, global and ethical impacts.

Most recent research looks at the possibilities of how play, thematic or project based learning, especially in the early years, can be used to help teach the concepts of computational thinking and coding. Coding experiences could be more like games than challenges, or more like pretend stories than problems to solve.

In the initial stages of design and development of the computing/technology curricula in most if not all of the jurisdictions, industry experts were asked for input and either sat on advisory panels or were part of the consultation processes. In some cases, large multinationals have produced, and are continuing to produce, a large amount of freely available resources for teachers to use aligned directly to local coding curricula.

Teacher qualification and awareness

In 2016 and 2017 in Finland, a survey was conducted as part of the government's analysis, assessment and research activities with teachers who had already accessed training on the use of technology in education. It found that, for most teachers, the practice of teaching coding was not familiar. Also, most teachers expressed being unable to teach coding and would like to have further training in coding and pedagogy concerning the subject.

During the consultation process for the new Digital Technology Curriculum for New Zealand (2017), it was reported that teachers wanted help to better understand how they would implement the new curriculum content. The New Zealand government's response was to provide funding of up to $40 million NZ dollars over the next three years, for continuing professional development of teachers before the new curriculum becomes mandatory in 2020.

In a recent report titled, *After the reboot: Computing education in UK schools* by the Royal Society, a number of recommendations were made regarding the teaching of the computing curriculum in England, Scotland and Wales.

- There should be a curriculum committee established to provide government with ongoing advice on the content, qualifications, pedagogy, and assessment methods for computing in schools.
- Government and industry need to play an active role in improving continuing professional development (CPD).
- Financial support should be made available to schools to release staff to attend professional development opportunities.

Conclusions

Analysis of the six curricula has shown that there is commonality in terms of what is taught at each age level and throughout each primary curriculum. Computational thinking and common coding standards, strands or themes can be clearly identified and are being implemented from the child's first year in primary school in all six selected countries.

In the countries that have computing or technology identified as a separate subject in the curriculum, all recommend that the implementation or teaching of coding or computational skills needs to be integrated throughout a number of primary curriculum subjects.

There has been an acceleration in the volume of international research on age-appropriate outcomes, guiding children's experiences of coding in primary school. However, practical investigation with schools is now required to understand and frame the potential benefits and challenges of implementing a playful, thematic or project-based learning approach to teaching code in Irish primary classrooms.

Continuing professional development of teachers in relation to code in primary schools is a priority in all six countries. As the world of technology is constantly changing there is a necessity for ongoing support for teachers in this area.

Glossary[10]

Abstraction: Abstraction is about simplifying things; identifying what is important without worrying too much about the detail. Abstraction allows us to manage complexity**.**

Algorithm[11]: An algorithm is a sequence of steps designed to accomplish a specific task. Algorithms are translated into programs, or code, to provide instructions for computing devices.

Beebots: A programmable floor robot for early years and lower primary.

Debugging: Errors in algorithms and code are called 'bugs', and the process of finding and fixing these is called 'debugging'.

Coding/computer programming: Coding and computer programming are often used interchangeably and refer to the process of developing and implementing instructions which enable a computer to perform tasks and solve problems. There are several different programming languages.

Computational Thinking: is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry out.

Computer science: Computer science is the study of how computers work. It involves the design and development of all types of software, from operating systems and phone apps, to interactive games and other forms of interactive technology.

Decomposition: The process of breaking down a problem into smaller manageable parts is known as decomposition. Decomposition helps us solve complex problems and manage large projects

Digital Literacy[12]: the basic skill or ability to use a computer confidently, safely and effectively, including: the ability to use office software such as word processors, email and presentation software, the ability to create and edit images, audio and video, and the ability to use a web browser and internet search engines.

[10] Glossary is adapted from the Oireachtas Library and Research Service: *STEM in schools: the introduction of Coding and Computer science/ICT to the curriculum*, 25 August 2017.

[11] The definition of 'algorithm' is adapted from that provided in the *NCCA Leaving Certificate Computer Science: Draft Specification for consultation*.

[12] The definition of 'digital literacy' is attributed to Simon Furber (2012) as cited in Keane and McInerney (2017).

Information and Communications Technology (ICT): ICT relates to using technology for access to information through communication. ICT includes a range of hardware and software devices and programmes such as personal computers, assistive technology, scanners, digital cameras, multimedia programmes, image editing software, database and spreadsheet programmes. It also includes communications equipment such as the Internet, email and video conferencing.

Logic: Logical reasoning helps us explain why something happens.

Patterns: By identifying patterns we can make predictions, create rules and solve more general problems.

Repetition: Repetition is the recurrence of actions or events.

Scratch: Pictorial block based coding language

Scratch Jnr: Pictorial block based coding language**,** specifically for children aged 4 -7ys.

Sequence: Sequence based algorithms are made from a precise set of instructions.

Variables: A variable is a simple way of storing one piece of information somewhere in the computer's memory whilst a program is running, and getting that information back later.

References

Bell, T and Duncan, C. (2015) *Proposed digital technologies curricula up to NZC level 5*. Cantubury Uni,

Berk, L.E., and Meyers, A.B. (2013). *The Role of make-believe play in the development of excutive function: Status of research and future directions.* American Journal of Play, 6(1), 98.

Bers, M. (2017) *Coding as a playground*. New York: Routledge.

Bers, M. U. (2012) *Design digital experiences for positive youth development: From playpen to playground.* Cary, NC: Oxford.

Coding/computer programming: European Schoolnet. (2015). Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe. Accessed on 14 September 2016 at http://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1- 8105-ea27c0d2bbe0

Computer science: Technology Schools.org. (2016). *What is the Difference Between Computer science and Computer Information Systems Majors*? Accessed on 24 July 2017 at http://technologyschools.org/what-is-thedifference-between-computer-science-and-computer-information-systems-majors/

Computing at School Scotland, http://www.cas.scot/

CSTA standards, Computer Science Teachers Association, *K-12 Computer Science Standards*, https://www.csteachers.org/page/standards

Department of Education (2013). *Computing programmes of study: key stages 1 and 2*, National curriculum in England, Reference: DFE-00171-2013.

Duncan, C., Bell, T. and Tanimoto, S. (2014). *Should your 8-year-old learn coding?* In Proceedings of the 9th Workshop in Primary and Secondary Computing Education on - WiPSCE '14, pages 60–69, New York, New York, USA, ACM Press.

Farrell, K. *et al.* (2017). *Teach Computing Science, A Guide for Early Years and Primary Practitioners.*

Finnish National Board of Education (2014). *National Core Curriculum for Basic Education*, The Scottish Informatics and Computer Science Alliance.

Johnson, J. S. and Newport, E. L. *Critical period effects in second language learning: The influence of maturational state on the acquisition of English as a second language*. Cognitive Psychology, 21(1):60–99, Jan. 1989.

Keane, N. and NcInerney, C. (2015) *Report on the Provision of Courses in Computer Science in Upper Second Level Education Internationally.* LERO, The Irish software research centre, report commissioned by the NCCA.

Koodi (2016), *Learning paths Finland*, http://koodiaapinen.fi/en/

Lee, I. (2012) *Computer Science: Critical K-8 Learning*. In P. Phillips, editor, Special Issue Computer Science K-8: Building a Strong Foundation, pages 10–11. Computer Science Teachers Association.

Margolis, J. and A. Fisher. (2013). *Unlocking the clubhouse: Women in computing*. MIT press.

National Council for Curriculum and Assessment (NCCA). *ICT (Information and Communications Technology) (2016)*. Accessed on 16 November 2016 at http://www.ncca.ie/en/curriculum_and_assessment/ict/

New Zealand Ministry of Education, (2017) *Digital Technologies Hangarau Matihiko, Draft for consultation*, http://www.education.govt.nz/ and www.tewhariki.tki.nz

Northern Ireland Council for the Curriculum, Examinations and Assessment, NI. http://ccea.org.uk

The Royal Society (2017). *After the reboot: computing education in UK schools,* royalsociety.org/computing-education.

Viteli, J. and Tanhua-Piiroinen, E. (2017). *Comprehensive Schools in the Digital Age 2017 - Status review and recommendations for action*. Tampere Research Center for Information and Media, TRIM, University of Tampere, Finland, 2017.

Wing, J. (2014). *Computational Thinking Benefits Society*. 40th Anniversary Blog of Social Issues in Computing. http://www.utad.pt/vPT/Area2/eventos/Documents/Artigo%203.pdf

NCCA