Computational Thinking

in the Curriculum

Keith Aquilina

Education Officer (Digital Literacy),
Directorate for Digital Literacy & Transversal Skills, MEDE, Malta

MA ODE, Dip CS Ed, B Ed (hons)

keith.aquilina@ilearn.edu.mt

Section A: Rationale

The vision of the directorate for Digital Literacy and Transveral skills is that of empowering learners to succeed as global digital citizens. This focus being on the adoption of 21st century competence (Digital Literacy paper). The how these may be adopted is through the development of digital competences both for students and for educators. This includes competence in Computational Thinking (CT) which experts in the field agree in seeing as an attitude and skill 'that should be discovered and exercised in a much greater way than it is' (Mohaghegh & McCauley, 2016). Bocconi et al (2016) point out that CT is a key 21st century competence in the context of compulsory education. CT encourages creativity, critical thinking and problem solving.

In September 2016 a Peer Learning Activity was organised in Helsinki as part of the workings of the Digital skills and competences group forming part of ET2020. This PLA focused on Coding and CT in the Curriculum. From this meeting the following main targets have been identified which could help a national systematic adoption of CT where it is introduced as a methodology to bridge the gap between current curricula and the needs of learners, society and industry preparing for a digital world;

1. Set an operational definition for CT that is meaningful to educators,
2. Develop CT learning outcomes as part of the digital literacy curriculum,
3. Identify sections within subject curricula where CT may be incorporated,
4. Elicit Grass Roots initiatives,

This is in line with the embedding of the Cross Curricular Themes which is suggested in the Learning Outcomes Framework document i.e. 1) embedding in the subject learning outcomes, 2) through the pedagogy approach and 3) through the school activities, events and policies

Section B: Definition

Throughout our lives we experience situations involving Critical Thinking, Problem Solving and Decision Making. This includes school experiences of learners where for example, they use critical thinking skills to plan and conduct research, manage projects whilst solving problems, and make informed decisions using appropriate digital tools and resources. CT encourages creativity, critical thinking, decision making and problem solving.

In the conclusions of the working group ET 2020, held in Helsinki in September 2016, it was pointed out that various definitions are used throughout the globe to explain CT. 'Some emphasise particular aspects of CT and or reflect specific stakeholder positions (Bocconi et al. , 2016). On a local level, most notably the differences emanate from the perspectives of various stakeholders and the backgrounds they are associated with. Whereas in the Malta ICT skills audit 2017 conducted by the eSkills Malta foundation the focus is on the problem-solving skills required by the industry, the directorate for digital literacy and transversal skills is more inclined at focusing at computational thinking as a transversal component in learning. Although these definitions at times are contrasting (Mohaghegh & McCauley, 2016), everyone seems to agree with Wing (2016) who outlines that CT 'represents a universally applicable attitude and skill'. CT can be referred to as a competence which DePaul (2009) describes as facilitating new ways of seeing existing problems, emphasizing creating knowledge rather than using information, presenting possibilities for creatively solving problems, and facilitating innovation. Similarly BBC (2017) mention that CT allows for the understanding of a complex problem and the development of possible solutions. Such solutions being meaningful for 'a computer, a human, or both'. The definition which has gained attention internationally and which locally fits well with the other dimensions of Digital Literacy as a Cross Curricular Theme is as follows;

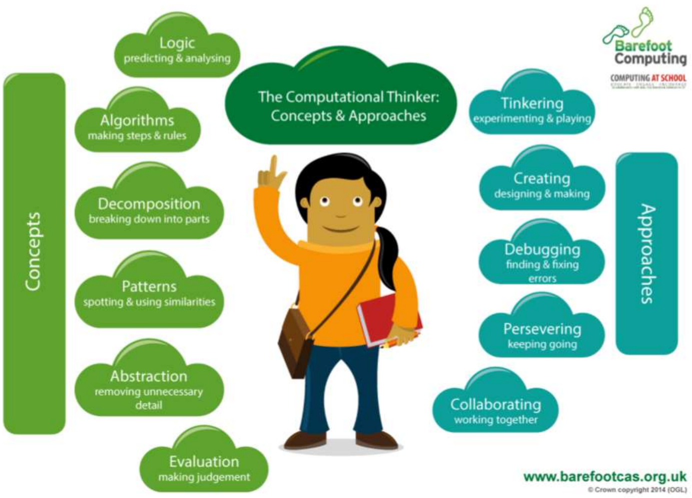
> 'Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can effectively be carried out by an information-processing agent.'
>
> Jeanette Wing (2010) Computational Thinking: What and Why?

Section C: Core Components of CT

An operational definition needs to be shaped allowing for better understanding of what is being sought; where such a definition needs to be meaningful to all educators. The identification of CT's core components may give such a definition a better meaning.

Mohaghegh & McCauley (2016) point out that CT 'involves a number of core principles from computer science, such as abstraction and algorithm design, decomposition, pattern matching, generalization, and inference.' Similarly, Google (2017) outline characteristics of which logical ordering and analyzing data and creating solutions using a series of ordered steps (or algorithms), and dispositions, such as the ability to confidently deal with complexity and open-ended problems.' BBC (2017) describe four key cornerstones to CT which are;

- decomposition - breaking down a complex problem or system into smaller, more manageable parts,
- pattern recognition – looking for similarities among and within problems,
- abstraction – focusing on the important information only, ignoring irrelevant detail,
- algorithms - developing a step-by-step solution to the problem, or the rules to follow to solve the problem.



Meanwhile Barefoot (2014) give a more indepth structure to the characteristics of CT where the following core concepts are identified;

- Logic – predicting and analysing
- Algorithms – making steps and rules
- Decomposition – breaking down into parts
- Patterns – spotting and using similarities
- Abstraction – removing unnecessary detail
- Evaluation – making judgement

Such core components give a better understanding of what CT really encompasses and thus better explain how and where it can relate to subjects thought in our schools. CT can be understood as directly linked to and as a component of digital learning and which in a learning situation may be enabled through approaches which Barefoot and other entities point towards tinkering, making, collaboration, communication and creative situations.

In the following sections, these core components will be delineated throughout the primary and secondary educational system, identifying learning outcomes and giving examples of how such concepts may be learnt through practical hands-on learning activities.

Section D: CT Learning Outcomes

The following is a list of definitions (see Barefoot, 2016) and generic learning outcomes, grouped by the core components identified in the previous section. These generic learning outcomes give a clear idea of what competence each learner is to achieve by the end of the obligatory scholastic cycle.

Core Concept	**Definition**	**Learning Outcome/s**
Logic predicting and analysing	Logic is the study of reasoning. The purpose of logic is to help us try and make sense of things: it helps us establish and check facts.	• I am able to predict and explain the outcomes of a sequence of instructions.
Algorithms making steps and rules	An algorithm is a sequence of instructions, or set of rules, for performing a task.	• I am able to create and modify a sequence of instructions which give a solution to a given task.
Decomposition breaking down into parts	Decomposition is breaking a problem or system down into its parts. It sometimes involves breaking those parts down further. Decomposition helps us solve complex problems and manage large projects.	• I am able to break down a complex task into small, meaningful parts.
Pattern recognition spotting and using similarities	Using patterns means spotting similarities and common differ-ences. By identifying patterns we	• I am able to combine a sequence of instructions to follow a pattern.

	can make predictions, create rules and solve more general problems. This is called generali-sation.	• I am able to find similarity between different things. • I can recognise and find patterns or trends. • I am able to generalize and transfer a problem solving process to a wide variety of tasks.
Abstraction removing unnecessary detail	Abstraction is simplifying things; identifying what is important without worrying too much about the detail thus managing complexity. Abstracting leads to a simple view of the main idea of a thing.	• I am able to ignore detail that is not of interest, simplifying a complex task. • I am able to represent data through abstractions such as models and simulations • I am able to analyse a solution to a task and formulate a more efficient solution.
Evaluation making judgement	Evaluation is concerned with making judgements, in an objective and systematic way whenever possible. Evaluation is something we do every day: we make judgements about what to do and what we think based on a range of factors	• I am able to identify and correct errors in a sequence of instructions. • I am able to analyse and evaluate a solution to a given task.

In appendix 2, the Curricular Outcomes table shows these learning outcomes broken down by level thus showing the degree of competence to be reached according to the age group of the learners. In the second table specific examples are given showing how such concepts may be learnt through practical hands-on learning activities. The learning activities will use approaches as suggested in the Barefoot (2016) approaches (i.e. Tinkering, Creating, Debugging, Persevering and Collaborating) and incorporated in the main stream subjects.

Section E: Practical Tasks & Teacher Training

Current practices involve teachers and specialist digital literacy support teachers popularising the area by doing one off sessions in schools or events. Various resources are ready available for class teachers. In primary there are BeeBots, Pro Bots, Lego WeDos and Coding apps on OTPC tablets. In secondary there are Lego Mindstorms kits. These provide a fertile ground through which CT learning may flourish.

Such initiatives although commendable, do not have any structure and do not follow set outcomes. ‘A deeper understanding of computational problem solving is more valuable than exploring the surface of tools in this area without realising their full potential.’ Mohaghegh M. & McCauley M., 2016. Thus it is imperative that apart the availability of resources, teachers are given the needed guidance and training in this area. As mentioned by Bocconi et al (2016), such training needs to be followed by classroom support and online support. Moreover mapping such activities to the CT learning outcomes will show any gaps which need to be worked upon.

Section F: Grass Root Events

Stakeholders both within and outside school communities need to realise the importance of computational thinking as part of other 21st century competences which prepare learners to be active digital citizens and lifelong learners. Thus the adoption of CT within the classroom will be truly possible. Case in point, current situation allows for specific lessons at K-6 to focus on coding. This is done as part of the Hour of Code and Code Week initiatives. Throughout Europe, various Grass Roots initiatives are set up in order to popularise computational thinking and coding. Such initiatives should be done through collaboration within MEDE (e.g. Curriculum EOs; Maths, Science, ICT, Computing) and through cooperation with other stakeholders e.g. with MCA, FES, MCAST, MITA, IEEE, private educational institutes and industry. Initiatives could including eliciting participation in;

- National events
  - EU Code Week
  - Hour of Code
  - Robotics Week
  - Digital Literacy Week
  - Malta Robotics Olympiad
- Initiatives
  - Family Code Night in schools (ISTE, 2016),
  - Code Clubs,
  - Open days in schools,
  - Mini Embeds
- Training for parents and educators; To equip teachers, whatever their level of confidence, with the knowledge and the materials they need to teach CT effectively,
- Resources for teaching CT to K-11 students,
- Talking points for getting administrators on board with teaching digital literacy at primary and secondary level.

At a directorate level, rather than owning these events, it is imperative that these are elicited through strategic partnerships and collaboration. Digital literacy needs to be expanded beyond one-time events. Grass root events inevitably have the aim of popularising this area, showing the benefits that such learning may involve.

Section G: Computational Thinking in the Curriculum

Terms such as coding, programming and CT are many times used interchangeably but surely pedagogy experts in the field underline fine differences that exists. Bocconi et al (2016) mention that CT is more than programming and the relationship with digital literacy might not be able to capture fully the core ideas and skills associated with CT. 'Coding/programming is a constituent of CT, in that it makes CT concepts concrete and can thus become a tool for learning' (Bocconi et al, 2016). García Peñalvo (2016) mentions that although coding is so interesting, it is more important to emphasize in the idea of computational thinking as the application of high level of abstraction and an algorithmic approach to solve any kind of problems.

Certainly, coding and programming support the teaching of CT, where it encompasses a wide range of abilities. This document relates CT to Digital Literacy as a Cross Curricular theme as identified in the NCF and LOF.

As mentioned by Wing (2014), CT is not just or all about computer science. The educational benefits of being able to think computationally—starting with the use of abstractions—enhance and reinforce intellectual skills, and thus can be transferred to any domain. 'Almost all disciplines have now been influenced by computational thinking in some way, in both the sciences and humanities.' (Mohaghegh & McCauley, 2016)

Establishing a shared understanding of what CT is and how it is contextualised may facilitate the process of curriculum integration.

At a primary level, from kindergarten to year6, students should be able to grasp the core concept such as sequencing, conditioning and pattern recognition. At a secondary level, as for digital literacy, a lot needs to be done to integrate in curricular subjects. Moreover subjects like ICT, Computing and IT VET play an important role in integrating CT concepts in their framework.

For this to become a reality a work plan (see appendix) has been set in December 2016 for calendar year 2017. This has been drafted in order to work systematically on the 4 main targets outlined in the first section. This will be followed with a thorough strategy with concrete targets in place. The area is still at its infancy, gaps are being identified and mitigated e.g. 'How is CT implemented and assessed in the classroom?' The main target is to provide an environment in which each learner may become a computational thinker.

Section H: Conclusion

Schools need to be supported in the adoption of CT. This needs to be done in a systematic approach involving all stakeholders. CT is not seen as a stand-alone subject but is integrated with other subjects (Maths and Science mainly, but other topics too). Coding is an important driver for CT as it can provide fun and engaging activities. Yet this should not be done at the cost of alienating from the nature of what CT is. We need to work with school leaders and with teachers to develop learning experiences which elicit CT. 'Science, society, and our economy will benefit from the discoveries and innovations produced by a workforce trained to think computationally.' Wing (2014)

References

Barefoot (2014), Computational Thinking, available online: http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking/

BBC (2017), Introduction to Computational Thinking, available online: http://www.bbc.co.uk/education/guides/zp92mp3/revision

Berry M. (2014), Computational Thinking in Primary Schools, available online: http://milesberry.net/2014/03/computational-thinking-in-primary-schools/

Bocconi S., Chioccariello A., Dettori G., Ferrari A. & Engelhardt K. (2016), Developing Computational Thinking in Compulsory Education: Implications for policy and practice, JRC Science for policy report. Available online: http://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188_computhinkreport.pdf

DePaul (2009), Computational thinking across the curriculum: A conceptual Framework, available online: http://compthink.cs.depaul.edu/Framework.pdf

García Peñalvo (2016) What Computational Thinking is, available online: http://repositorio.grial.eu/bitstream/grial/679/1/CT.pdf

Google (2017), Exploring Computational Thinking, available online: https://edu.google.com/resources/programs/exploring-computational-thinking/

ISTE (2016), How to Bring "Family Code Night" to Your Elementary School, available online: https://www.iste.org/resources/product?id=3990&name=How+to+Bring+%22Family+Code+Night%22+to+Your+Elementary+School

Mohaghegh M. & McCauley M., 2016, Computational Thinking: The Skill Set of the 21st Century, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (3) , 20, available online: https://www.researchgate.net/publication/303792583_Computational_Thinking_The_Skill_Set_of_the_21st_Century

Wing J.M. (2010), Computational Thinking: What and Why?, available online: https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Wing, J.M. (2014), Computational Thinking Benefits Society, available online: http://socialissues.cs.toronto.edu/index.html%3Fp=279.html

Wing J.M. (2016), Computational Thinking, available from: https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf

Wolfram, S. (2016), How to Teach Computational Thinking, available online: http://blog.wolfram.com/2016/09/07/how-to-teach-computational-thinking/

Committee for the Workshops on Computational Thinking; National Research Council (2011), Report of a Workshop of Pedagogical Aspects of Computational Thinking http://www.nap.edu/catalog.php?record_id=13170

Appendix 1: Work Plan 2017

The following work plan has been drafted in order to work systematically on the 4 main targets outlined in the first section.

Quarter 1, 2017

- Set concept document outlining working definition and objectives to be reached by years 3, 6, 8, 10,
- Discuss and gather feedback re definition and objectives; internally, Faculty of Education, EO ICT/Computing & HoDs,
- Seek collaboration with external entities,
- Update draft document and share internally,
- Feedback session / Workshop for eLearning support teachers
    a. Intro: outline definition and key concepts,
    b. Hands on stations with set activities per concept,

Quarter 2, 2017

- Include exemplar activity per LO / year group,
- Feedback sessions for volunteering class teachers,
    a. Intro: outline definition and key concepts,
    b. Hands on stations with set activities per concept,
    c. Create new activities for existing resources for specific years including for Tablet Year 4 apps,
- Issue first draft for consultation with curriculum EOs,
- Agreement/s in collaboration in this area,

Quarter 3, 2017

- Map with LOF subjects,
- Teacher training pilot program,
- Curriculum subject mapping.

Quarter 4, 2017

- Educator seminar to raise awareness, update via feedback,
- School PDs
- EU Code week,
- Hour of Code

Other items to consider;

a) Analysis of teachers' perception of students competence re CT
b) Analysis of students competence re CT
c) Develop teachers' understanding of CT pedagogy
d) Code Clubs
e) Provide a toolkit for schools with
    - Curriculum
    - Resources by year
    - Professional Development session plan & resources

Appendix 2: Curricular Outcomes

In the following table, the learning outcomes identified in the previous section are split

KG- year 3	4 - 6	7 - 8
Objective: Logic		
Predict the outcome of a linear program Create and debug simple programs	Explain what a simple algorithm does. To create, edit and refine sequences of instructions for a variety of programmable devices.	
Objective: Algorithms		
Design programs that accomplish specific goals.	Explore different ways in which software can be planned. To begin to write simple linear scripts.	Write complex scripts involving conditioning.
Objective: Decomposition		
Solve a problem by breaking down into smaller parts.	Breaking down a problem into smaller, manageable parts	Solve a complex problem by breaking down into smaller meaningful parts.
Objective: Patterns		

Combine a sequence of instructions to follow a pattern or create a shape.	Use repetition in programs	
Objective: Abstraction		
I am able to ignore detail that is not of interest.	Abstraction Reducing complexity to define main idea	
Objective: Evaluation		
	Detect and correct errors in algorithms. Debug programs that accomplish specific goals.	

Appendix 3: Exemplar Activities

The following activities have been compiled through extensive online research (e.g. Barefoot and BBC) and in-house discussions. These are intended to give an insight on how students, at different levels, may be introduced to each CT core concept. As mentioned in the previous section, the focus on approaches such as Tinkering and Making.

KG- year 3	4 - 6	7 - 8
Logic		
Learners might predict the outcome of Bee-Bot programs.	Learners may use logical reasoning to explain what their algorithms or code does or to help them debug Scratch programs	
Algorithms		
Learners might predict the outcome of Bee-Bot programs.	Learners may use logical reasoning to explain what their algorithms or code does or to help them debug Scratch programs	
Decomposition		

Create directions to a location in the school by breaking the directions down into smaller geographical zones. Join the sections of directions together into a whole. Learners might predict the outcome of Bee-Bot programs.	Develop a plan to make the school "green." Separate strategies such as recycling paper and cans, reducing use of electricity, and composting food waste. In planning the publication of a monthly newsletter, identify roles, responsibilities, timeline, and resources needed to complete the project. Learners may use logical reasoning to explain what their algorithms or code does. Learnings may reason help them debug simple programs such as Scratch programs.	Consider the large-scale problem: "What does it take to become a rock star?" Break it into smaller parts. Discuss what variables are within a student's control and what variables are determined by outside factors.
Patterns		
Learners might tinker with a Bee-Bot to find out what happens, as they do this they notice what the Bee-Bot does as they press the different command keys; they build up rules about how the programming language works.	When designing a game, the learner might compare a number of example games to work out the common features.	In Chemistry, learners might determine the rules for Chemical bonding and interactions.

Objective: Abstraction		
https://www.youtube.com/watch?v=jV-7Hy-PF2Q With many sizes and colors of three-sided shapes, the abstract is a triangle. Learners might summarise the action for a simple animation of a joke as a storyboard.	Hear a story, reflect on main items, and determine an appropriate title. After studying a period in history, identify symbols, themes, events, key people, and values that are most representative of the time period (e.g., coat of arms). When creating a computer game the learner might model how a ball bounces, but not the effect of air resistance	Choose a period in politics that was most like the current one by analyzing the essential characteristics of the current period.
Objective: Evaluation		
Learners might think about how their work could be improved	Learners might consider the views of others to improve their work. Learners can evaluate finished work and also make evaluative judgements as they work on a project.	Learners can evaluate finished work and also make evaluative judgements as they work on a project. Learners may remix programs such as Scratch projects.