

Think before you ... code!

Objective 1.6 of the Digital Strategy for Schools is to provide opportunities for students to pursue in-depth ICT study in the senior cycle. This senior cycle subject to include elements of coding is set to be offered to Irish students from September 2018. *Developing Computational Thinking in Compulsory Education – Implications for Policy and Practice* was released by the EU's Joint Research Centre in December 2016 and acknowledges that successful computational thinking (CT) integration in compulsory education faces unresolved issues and challenges.

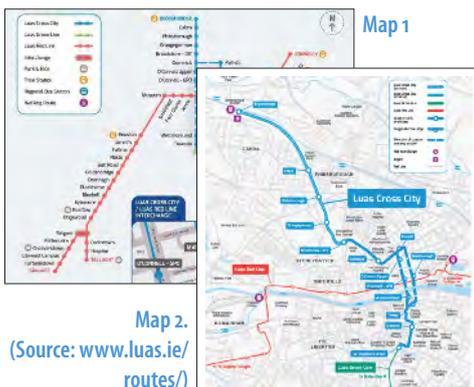
As primary teachers, we build the foundations of knowledge and skills for a variety of subject areas but the foundations for computer science are not merely coding – but computational thinking whereby "... CT is a problem-solving methodology that expands the realm of computer science into all disciplines ..." (p.6). Appreciating that CT is a problem-solving methodology, it is good to draw on a bank of challenges for our pupils:

- ▄ *Bebras Challenge* at bebras.org/ has a huge variety of problems which can be solved using pen/paper or computers but all draw on CT skills.
- ▄ *CS Unplugged* at csunplugged.org/ is a collection of challenges that use cards, string, crayons and lots of running around – it is not the rote learning of lines of code!

Core computational thinking skills

Abstraction: "...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..." (Csizmadia et al., 2015, p. 7).

Public transport on the Luas is a good example. Map 1 is a very good representation of the order of the stops on the various lines. Map 2 is a better representation if we want to see the



location of the stops and how far they are from each other. Learning what information is important and what can be left out is a critical skill for pupils to develop as problems grow in complexity.

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) thus pupils demonstrate algorithmic thinking when they create or use a well-defined series of steps.

Challenge: what are all the prime numbers from one to 100? (i.e. Sieve of Eratosthenes used to explore prime numbers)

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

1. Two is a prime number because it can only be divided evenly by itself and one.
2. Cross out all numbers two will go evenly into.
3. Three is a prime number because it can only be divided evenly by itself and one.
4. Cross out all numbers three will go evenly into etc.

When can you stop? Defend your answer!

Decomposition: Decomposition means breaking down a complicated problem into its components and working on one component at a time – done to good effect in PE as we break down sequences of movement in athletics, dance etc. into related steps.

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). If pupils are set a challenge to plot the fastest way home, they can test a number of solutions by varying conditions (walking/cycling etc) drawing on the skill of debugging).

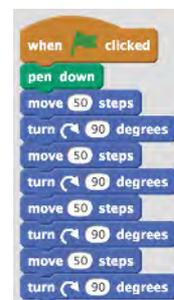
Pattern recognition: Asking questions such as 'Is this similar to a problem I've already solved?' and 'How is it different?' are important here, as is the process of recognising patterns both in the data being used and the processes/strategies being used. (Csizmadia et al., 2015, p. 8).

A challenge to draw regular polygons

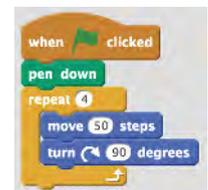
can lead to pattern recognition. To draw a square on the playground surface, the algorithm might look like the following:

1. Draw one side of the square.
2. What angle do we need to turn before we draw the next side?
3. Draw the next side etc.

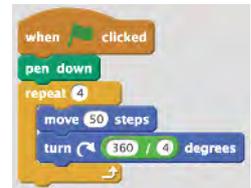
This algorithm can be applied to the drawing of any regular polygon with the angle of rotation being the only variable (if you keep the length of the sides the same). A pattern/generalisation will emerge with experimentation and trial/error. Transferring the problem to *Scratch* (visual programming language for children at scratch.mit.edu/) the possible algorithms and patterns emerge.



1. Algorithm to draw a square



2. More concise...



3. The link between number of sides and angle of rotation i.e. angle of rotation = $360/\text{number of sides}$

Of equal importance are the dispositions/attitudes/attributes that can be developed through CT – i.e. confidence in dealing with complexity, persistence in dealing with difficult problems, the ability to handle ambiguity, the ability to deal with open-ended problems as well as the ability to communicate and work with others to achieve a common goal or solution (Barr, Harrison & Conery (2011, p. 51)

The challenge for us as teachers is to have a shared understanding of CT (appreciating it is much more than coding) and it is not confined to any one subject area. Of further importance is the provision of meaningful support to teachers to embed CT in teaching and learning drawing on ongoing research and practise in Ireland and internationally for the holistic development of Irish pupils.

Written by SEÁN GALLAGHER,
Professional Development
Service for Teachers (PDST)

