SYSTEMATIC COMPUTATIONAL THINKING DEVELOPMENT: A GUIDE FOR TEACHERS AND LEARNERS

Noha Abdulkhalig M Alharbi

Noha.alharbi.2@research.gla.ac.uk

Under the Supervision of:
Prof. Quintin Cutts

Quintin.Cutts@glasgow.ac.uk

2021

Executive Summary

Computing is one of the most important skill in everyday life. The teaching of computing skills should be studied carefully, because:

1. It has particular aspects of learning.
2. Anyone can learn it.
3. It should start from early age
4. It has real world aspect for engagement.

Historically, computing skills have often been treated as innate and learners can learn them or they cannot.

Recent research efforts show that computing is a skill that can be learned by anyone by adopting specific learning methodologies.

Just as in other subjects like maths and literacy, we should start it early. Learning material that presents this view can play a major role in engaging young learners. Learning materials and teaching approaches should promote real-world problem-solving and ensure that learners can see themselves represented through computing.

In this short document, we outline key elements to develop and enhance Computational Thinking (CT) ability among learners through a clear presentation of the skills introduced and responsive teaching approaches, and we provide guidelines on how to incorporate them into computing curricula. The guidance also encourages teachers to evaluate existing resources and offers advice on how to adapt lessons to support their own students, along with examples of best practice.

We know that many primary teachers don't know a lot about computing. But we also know that you know more than computing experts about teaching young people. So we want to work collaboratively with you to find the best solution to enable all our young people to succeed and to ensure you have a rewarding teaching experience.

Introduction

What is meant by 'computational thinking'?

Computational Thinking (CT), a term used since the 1970s, involves structured thinking methods to develop processes or procedures for real world tools and problem so that computers can produce appropriate output for given inputs. It is the focus for students' knowledge development about designing computational solutions to problems, algorithmic thinking, and coding.

During the last years, there has been an increasing interest about CT education in K-12 schools, and its role in children's acquisition of thinking skills and digital competencies. It turns out to be useful for human problem solving too – so it is learning of value to every child, no matter their destination.

It is well accepted in the literature that CT involves a number of skills, like problem decomposition (breaking down complex problems to simpler ones), developing

algorithms (step-by-step solutions to problems), and abstraction. But to be a good computational thinker (creating solutions of use in many situations) requires more than this, and we present a single framework here for understanding the full range of requirements.

Motivation

One of the major motivations behind teaching computer science is improving problem-solving skills among learners. There are two types of problem solving; 1) interim problem that come across as part of subject related problem e.g. some problem that is to solve as Quantum Physics part and 2) real world or every day problem solving.

We are interested in real-world problem solving that can be achieved by teaching certain aspects of computer science. Such real-world problem solving is often not well-addressed in other subjects. For example, maths focusses on so called “drill and practice” problem solving. However, the methodology adopted for the teaching is crucial if we are to gain the desired results.

Learning Curve

High level first or related to what learners already know about problem solving in the real world. For any problem/task, there are two domains of interest. For example, consider the problem that I need to go from point A to B. The first domain is the Problem domain that is the context in which the problem is set. In the domain of moving from A to B, I know about location, geography, time and movement. The second domain is the solution domain. For this problem, this the domain of interest includes travel mechanisms that I know about: walking and where I can walk; cycling and cycling paths; cars and roads and driving; buses and trains and their routes and timetables and how they operate. With these two domains in place, I can consider Problem solving: analyse what I know about A to B; pick tools and technologies; develop sequence of steps that use the chosen tools to get from A to B.

Given this way of considering real world problem solving, the process of teaching a concept is composed of three main areas. These areas are problem domains, computing domains and problem solving domain with each spiral cycling through the three areas. In spiral teaching approach, the next concept is built on the previous concept. This is how the whole subject is taught. The concepts are inter-related.

The real world problem solving can be explained through real world example, Consider the problem solving in literacy for creative writing about a tree. First phase is understanding the problem domain. In this phase, the learner should know what is meant by tree, branches, leaves and flowers etc. Learners learn this at a very early age as they are exposed to the world if they live in a country with trees. Second phase is solution domain where the learner should be knowing the language in which he has to describe the tree. It includes know the

syntax and grammar rules of the language. The language could be any language such as English or Arabic. This knowledge comes first from hearing parents and other friends and relations talking about the world and then later in learning to reading. All this helps to acquaint the learners with the tools and rules of their language. The third area is about problem solving in which he has to describe the tree using the chosen language. In this he will need to write descriptions, using the tools of the language, write describe the tree in the given language, matching to his understanding of what a tree is.

Figure 1: CS Learning Curve

Scratch Programming Tool

Scratch is a free visual coding tool. It is a type of “block programming which uses graphical blocks to represent program commands” (Crook, 2009). Students can use Scratch to “code their own interactive stories, animations, and games. The Graphical User Interface (GUI) of Scratch tool is shown as follows:

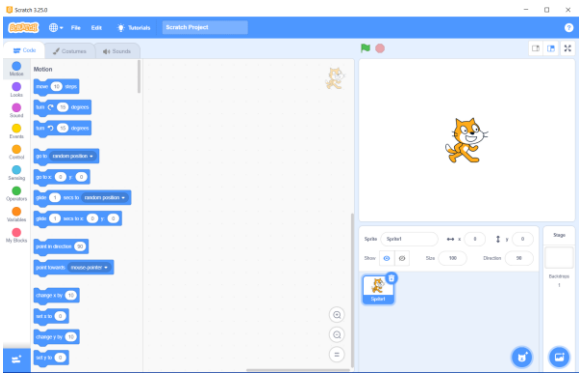
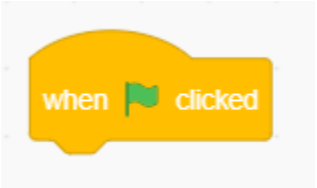


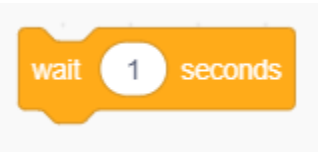
Figure 2: Scratch tool GUI

Sprite: Start with teaching about Sprite and how these works. Sprite are the objects that show visual behaviour for the given instructions. The objects available in Scratch tool and can be downloaded from Internet as well. The code always work behind a sprite. Cat is an example of Sprite.

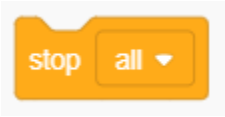
Program Starting Point: It is the instruction to computer in visual form to start the program. In Scratch, the instruction is as follows:



The next introduction to block used for describing sequences.



Program Ending Statement: A program instruction to end the program



Backdrop: these are background images that are available in Scratch tool or can be selected from the computer.

Start with Example (Find Path from Home to School)

This is the example explaining the problem solving through computing process. We explain the real world processes in computing terms.

Problem Statement:A driverless car has to go from home to school on street map.

Sequences

Stage 1: Understanding the problem domain

The learners need to understand the problem and different related concepts. He should also knowing the problem context. For example, they should be knowing what is school, what is a map, streets and different other objects on map etc.

- Given Inputs: A Map, A car, School and home Icons
- Given Resources: A4 Paper, Color Pencils
- Expected Output: Directions from home to school

Stage 2: Learning of Computing Tools

Programming language is set of instruction that tell how to match from problem domain to solution domain. Different programming tools can be used to problem solving. However, Scratch tool is selected to different computing processes in this guide because it comprehensively covers these processes. Scratch tool is taught for teaching
sequences.

Stage 3: Problem solving

The problem solving is three step process. First step is problem analysis in which learner with understand the problem in computing terms. It will identify that the car has to go on sequence of turns on path to reach destination. For this he needs a map to understand the paths, different objects (gardens, petrol station, market etc) on map. Second step is computing domain in which he will understand how the Scratch tool can be used to write the program and third step is implementing using Scratch in which he will perform following steps:

1. Go Straight (S)
2. Road Blocked. Go Left
3. Go Right
4. Go Right

5. Go Left and School reached.

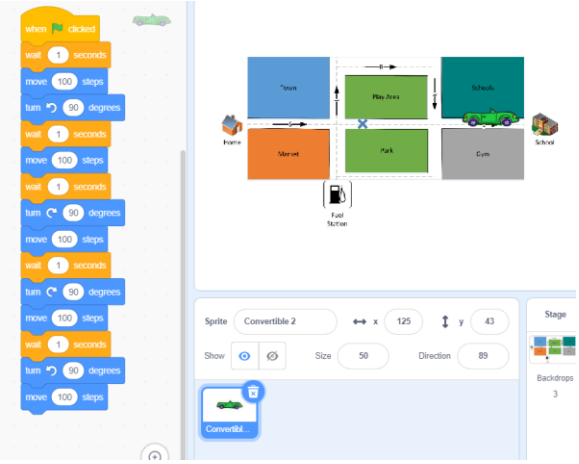


Figure 3: Example of Simple Sequences

Conditions

Problem Statement: A driverless car has to go from home to school in map after taking fuel from gas station

Stage 1: Understanding the Problem

The learners need to understand the problem and different related concepts. For example, they should be knowing what is school, what is a map, streets and different other objects on map etc. However, they should also be knowing that there exist a condition of going to go to fuel station.

- Given Inputs: A Map, A car, School and home Icons
- Given Resources: A4 Paper, Color Pencils
- Expected Output: Directions from home to school

Stage 2: Learning of Computing Tools

Scratch tool (sequences + conditions)

Stage 3: Problem solving

1. Start from School Straight (S)
2. If to take gas for car,
    a. Go Right (R)
    b. Take fuel
    c. Go Left (L)
    d. Go Left (L)
3. Else Go Straight (S)
4. Go Right (R) and School reached

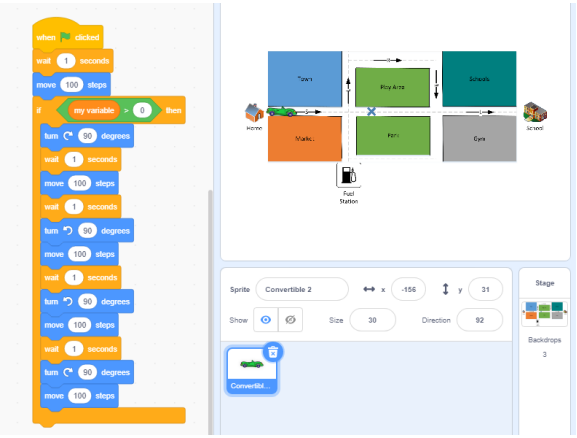


Figure 4: Example of Sequences and Conditions

Iterations

Problem Statement: A driverless car has to go from home to school in map after taking fuel from gas station and 5 round trips of Park.

Stage 1: Understanding the problem

The learner should understand the problem context and given inputs, processes and expected outputs.

- Given Inputs: A Map, A car, School and home Icons
- Given Resources: A4 Paper, Color Pencils
- Expected Output: Directions from home to school

Stage 2: Learning of computing tools

Scratch tool (sequences + conditions + iterations)

Stage 3: Problem solving

1. Start from School Straight (S)
2. If to take fuel for car
    a. Go Right (R)
    b. Take fuel
    c. Take 5 rounds of Park (L-L-L-L)
    d. Go Left (L)
    e. Go Left (L)
3. Else Go Straight (S)
4. Go Right (R) and School reached

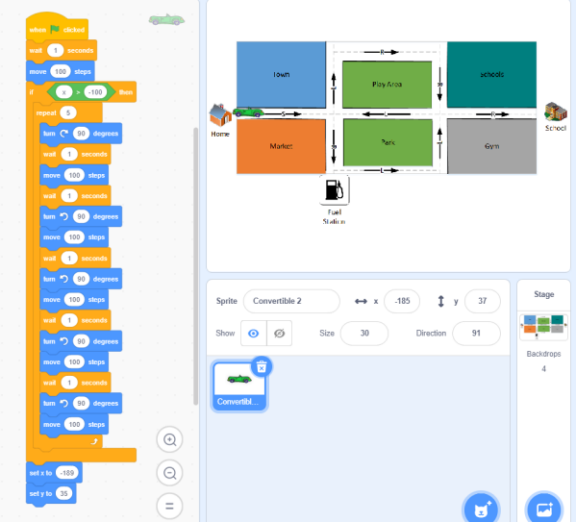


Figure 5: Example of Sequences, Conditions and Iterations

Justification

The structuring of teaching is adopted to develop inter-related concepts among learners. The examples are provided to the teacher to demonstrate the teaching methodology of spiral mode

Curriculum Example

Example 1: Making a Sandwich

Problem Statement: The problem is to make an animation of sandwich making process.

Understanding a Problem Domain

Learner should know cheese, bread, toast and timing. He should also know the sandwich making process and the basic principles of animation.

Computing and Modelling Domain

Teach basics of Scratch programming language. Give concepts of Sprite, Variables, Blocks, Starting point and Ending point. Explain these concepts on multimedia and focus of sequential instructions (to follow up in third stage for sandwich making).

The model to explain the sandwich process in computing terms in the Scratch describes the sequence of steps that involved in the process.

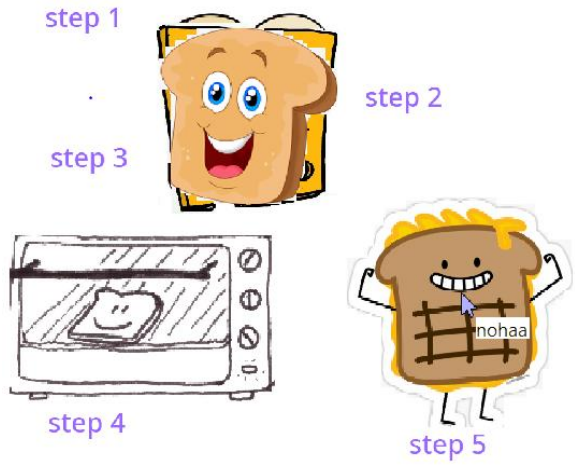


Figure 6: An animation explaining steps of sandwich making

Problem Solving

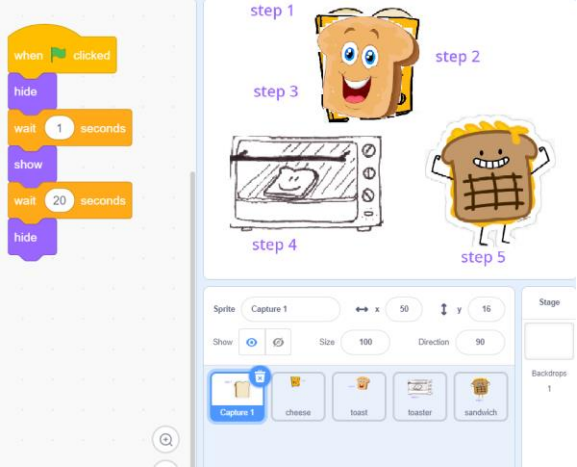
This is three stage process. Now the learners will solve the problem given in the first stage. First step is problem analysis and understanding in which learner will identify the inputs, processes and outputs using the domain knowledge. The output of this phase is as follows:

1. Inputs: Two pieces of bread and some cheese
2. Process: apply the bread pieces and cheese in sequential order, put in sandwich maker, power up and wait until sandwich gets cooked.
3. A brown sandwich

Second is writing algorithm in plain English/Arabic explaining the steps point by point. An explanation of algorithm is as follows:

1. Input = two bread pieces
2. Input = Cheese
3. Output = Brown Sandwich animation
4. Apply cheese on one bread piece
5. Put another bread piece on it
6. Put these in toaster and apply timer for 20 seconds
7. Get the Brown Sandwich

Third step is to apply these sequences in Scratch tool.



Fourth step is testing and check if the actual output is desired output.

Example 2: Brush your Teeth

Problem Statement: Problem is to brush the teeth.

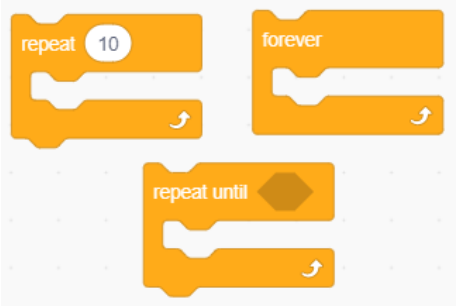
Understanding a Problem Domain

Learner should know about teeth, a brush and toothpaste. Learner should know the teeth cleaning process.

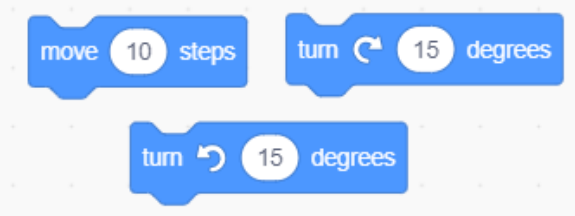
Computing and Modelling Domain

Teach basics of Scratch programming language. Give concepts of loading sprites from computer, iterations and movement related constructs to teach as follows:

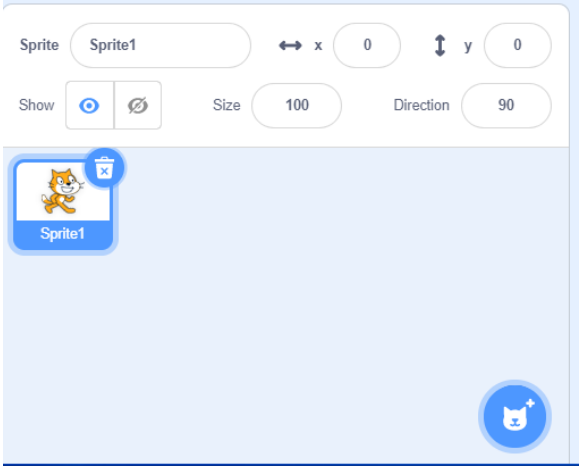
Iteration: repeating a task for given number of time. It is explained as Loops in Scratch.



Movement: it refers to movement of sprite for some step and in certain direction.



Load Sprite: load the images from local computer and set its properties.



The model of tooth-brushing process in terms of computing process described in Scratch tool is given as follows:

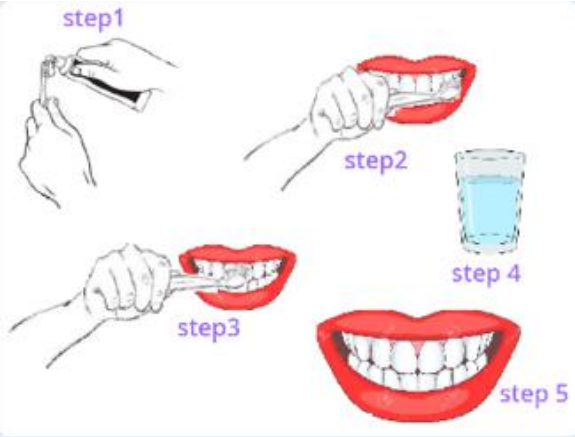


Figure 7: Model explaining steps of brushing teeth

Problem Solving

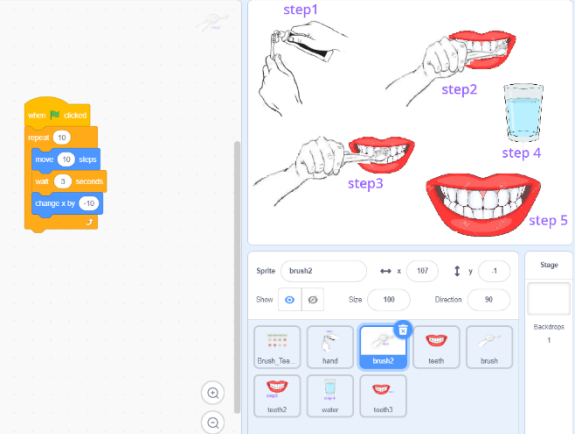
First step is problem analysis and understanding in which learner will identify the inputs, processes and outputs. The output of this phase is as follows:

1. Inputs: Toothpaste, Tooth brush and glass of water.
2. Process: apply the tooth paste on toothbrush
3. Repeat the movement of brush on teeth 20 times up and down
4. Clean teeth with water.

Second is writing algorithm in plain English/Arabic explaining the steps point by point. An explanation of algorithm is as follows:

1. Input = tooth brush, tooth paste and water glass
2. Output = Clean teeth
3. Put tooth paste on tooth brush
4. Move tooth brush up and down 20 times
5. Wash teeth with water

Third step is to apply these iterations in Scratch tool.



Example 3: Cat go to park

Problem Statement: The cat has to reach from home to park after crossing the road and taking 20 steps towards park.

Understanding a Problem Domain

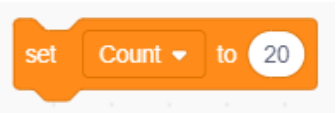
Going from home to park is daily life activity. An image map explaining the paths and different objects such as gardens, markets and hospitals is required. Problem is to reach coffee shop in 40 steps but do not cross the road if bus is coming.

Computing and Modelling Domain

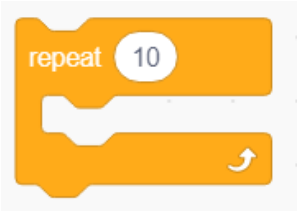
Teach basics of Scratch programming give concepts of loading background, iterations and conditions. Sequence of instructions is also told as part of learning.

Background: the image background for the code. It can either be loaded from local computer or from Internet.

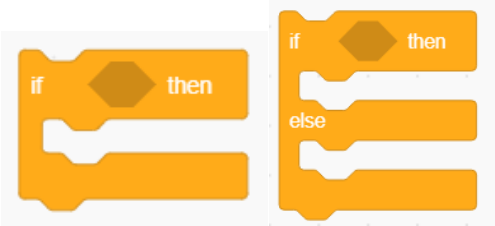
Variable: These are placeholders of values. For example, count is a variable with set value of 20.



Iterations: These are repeated condition that have to follow fixed number of time.



Conditions: these are statements used to make a decision on given condition.



Sequences: sequence of instructions including iteration and conditional statements to perform the task.

Figure 7: Model explaining steps of cat fun

Problem Solving

First step is problem analysis and understanding in which learner will identify the inputs, processes and outputs. The output of this phase is as follows:

1. Inputs: A maps explaining bus, cat, park, road and coffee shop.
2. Process: check if bus coming then stop else move 20 steps straight and then take right turn and again take 20 steps. Cat will reach the park.
3. Cat in park

Second is writing algorithm in plain English/Arabic explaining the steps point by point. An explanation of algorithm is as follows:

1. Input = A map showing roads, park and signals, cat, bus
2. Output = Cat at park
3. Cat is on footpath
4. If Bus Not coming
5. Move 100 steps
6. Turn right
7. Move 100 steps
8. Else
9. Stop.
10. Cat is in park

Third step is to apply these iterations in Scratch tool.