Developing creativity and computational thinking in your computing classroom

Developed by:
Mark Dorling
mark.dorling@computingatschool.org.uk / @MarkDorling
National CPD Coordinator
Computing At School

Concepts and structure were contributed to by:
Thomas Ng
CAS Board of Management
School Improvement Adviser (ICT & Assessment)
West Berkshire Council

Computing At School is supported and endorsed by:
Aim

• Develop a shared understanding of computational thinking
• Be able to share that understanding of computational thinking with your colleagues in school
• Know how to plan learning opportunities underpinned by computational thinking
• Identify computational thinking when observing lessons
• Understand why teach computational thinking
National Curriculum purpose

Computational thinking sits at the heart of the national curriculum programme of study for Computing. The opening sentence states:

- “A high quality computing education equips pupils to use computational thinking and creativity to understand and change the world”

(DfE 2013b, p. 188).
Your thought: Curriculum?

• “Curriculum is the contents of a course.”

• “A complete programme of learning or structured teaching plan for a course.”

• Curriculum refers to the means and materials with which students will interact for the purpose of achieving identified educational outcomes.


• Curriculum “means the planned interaction of students with instructional content, materials, resources, and processes for evaluating the attainment of educational objectives.”

http://www.doe.in.gov/asap/definitions.html Last Accessed: 23rd October 2010

• Curriculum is “an integrated course of academic studies.”

Your thoughts: Creativity?

• “Creativity is having the ability to create; original, expressive and imaginative work”.

• “Creativity is the capacity to produce something which is both unique and useful.”
  

• Generates and/or recognises how best practice and imaginative ideas can be applied to different situations.
  
  www.ucas.ac.uk/seps/glossary  Last Accessed: 23rd October 2010

• “Creativity is a mental process involving the discovery of new ideas or concepts, or new associations of the existing ideas or concepts, fueled by the process of either conscious or unconscious insight.”
  
Your thoughts: Computing?

• “the use or operation of computers”.

• “Computing is the study of how computers and computer systems work and how they are constructed and programmed.”

• Most individuals use some form of computing every day whether they realize it or not. The process of utilizing computer technology to complete a task. Computing may involve computer hardware and/or software, but must involve some form of a computer system.

• “Computing is any goal-oriented activity requiring, benefiting from, or creating algorithmic processes - e.g. through computers. Computing includes designing, developing and building hardware and software systems; processing, structuring, and managing various kinds of information; doing scientific research on and with computers; making computer systems behave intelligently; and creating and using communications and entertainment media.”
Your thought: Computational Thinking?

• *Do you know what it is?*

• *Do you know what it is NOT?*

• *Can you clearly define it?*

• *Do you know how computational thinking fits into Computing?*

• *Do you have techniques for developing in your lessons?*

• *Can you evidence or assess it in your classroom?*
Understanding National Curriculum
Understanding the breath & depth
CT in the Programme of study

• At KS1 (ages 5-7) pupils should be able to “understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.” (DfE 2013, p. 189)

• At KS2 (age 7-11) pupils should be able to (amongst other things): “solve problems by decomposing them into smaller parts” and also “use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs” (DfE 2013, p. 189)

• By the next key stage (ages 11-14) pupils should be able to: “design, use and evaluate computational abstractions …” and “use logical reasoning to compare the utility of alternative algorithms for the same problem” (DfE 2013, p. 190)
Summary: Computational Thinking

- **Computational thinking is:**
  - A way humans solve problems, not trying to get humans to think like computers
  - Conceptualising a problem, not computer programming (although it can be the end product).

- **It helps us to:**
  - Solve problems and design systems that none of us would think possible.
  - Consider what can humans do better than computers?
  - Consider what can computers do better than humans?
  - Consider what is computable?
Summary: Computational Thinking

• Why is it important in students in the classroom:
  - Confidence in dealing with complexity
  - Persistence in working with difficult problems
  - Tolerance for ambiguity
  - The ability to deal with open ended problems
  - The ability to communicate and work with others to achieve a common goal or solution

• It is a framework not a recipe to:
  – It’s a thought process to ask good questions
  – Characterising and understanding a problem
  – Guide computation design

• What does it look like in the classroom?
  – Student (attendee) lead activities / enquiry based learning

• Use the CAS computational thinking framework:
  - Developing computational thinking in the classroom the framework:
    - Stage 1: Definition
    - Stage 2: Concepts
    - Stage 3: Classroom techniques
    - Stage 4: Assessment

(CSTA 2014,204)
https://csta.acm.org/Curriculum/sub/CurrFiles/LLCTArticle.pdf
Stage 1: Definition for CT

• Computational thinking is:

“... the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent”

(Cuny, Snyder, Wing, 2010, cited in Wing 2011, p.20)

Wing indicates that these solutions can be carried out by any processing agent, whether human, computer or a combination of both

(Wing 2006)
Stage 2: Concepts of CT

Abstraction  
* e.g. identifying both the information that is necessary and the information that is not necessary

Decomposition  
* e.g. breaking problems and solutions down

Algorithmic thinking  
* e.g. defining solutions in terms of precise, well-defined and unambiguous rules

Evaluation  
* e.g. judging against external/objective criteria

Generalization  
* e.g. identifying commonality and re-using solutions or parts of solutions
Knowing the concepts?
Stage 3: Classroom techniques

Algorithmic thinking

- Identifying information not required for a solution
- Hiding functional complexity
- Hiding complexity in organization of data
- Identifying relationships between functional layers
- Identifying relationships between data abstractions
- Unpicking a generalized solution to identify layers
- Identifying information necessary for a solution

Decomposition

- Breaking down problems into sub-problems
- Breaking down solutions into sub-solutions
- Breaking down abstractions into layers of abstractions
- Organizing data into structures for processing
- Identifying relationships between functional layers
- Identifying relationships between data abstractions

Algorithmic thinking

- Expressing logic in a standard notation
- Sequencing
- Iteration
- Selection
- Procedural and functional thinking
- Parallel thinking
- Recursive thinking
- Unpicking a generalized solution to identify layers

Evaluation

- Making judgements based on external criteria
- Making trade-offs between competing requirements
- Making judgements about effectiveness (When is good enough good enough?)
- Making comparisons and reaching conclusions
- Interpretation of test results

Generalization

- Recognizing patterns in data
- Recognizing patterns in solutions
- Moving solutions or parts of solutions to different problems
- Identifying commonality

(Curzon, Dorling, Ng, Selby & Woollard 2014)
Evidencing CT in your classroom?
**Stage 4: Mapping evidence of CT to Progression Pathways**

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Programming &amp; Development</th>
<th>Data &amp; Data Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Shows an awareness of tasks best completed by humans or computers. (EV)</td>
<td>• Understands the difference between, and appropriately uses if and if, then and else statements. (AL)</td>
<td>• Performs more complex searches for information e.g. using Boolean and relational operators. (AL) (GE) (EV)</td>
</tr>
<tr>
<td>• Designs solutions by decomposing a problem and creates a sub-solution for each of these parts. (DE) (AL) (AB)</td>
<td>• Uses a variable and relational operators within a loop to govern termination. (AL) (GE)</td>
<td></td>
</tr>
<tr>
<td>• Recognises that different solutions exist for the same problem. (AL) (AB)</td>
<td>• Designs, writes and debugs modular programs using procedures. (AL) (DE) (AB) (GE)</td>
<td>• Analyses and evaluates data and information, and recognises that poor quality data leads to unreliable results, and inaccurate conclusions. (AL) (EV)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hardware &amp; Processing</th>
<th>Communication &amp; Networks</th>
<th>Information Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Understands why and when computers are used. (EV)</td>
<td>• Understands how to effectively use search engines, and knows how search results are selected, including that search engines use 'web crawler programs'. (AB) (GE) (EV)</td>
<td>• Makes judgements about digital content when evaluating and repurposing it for a given audience. (EV) (GE)</td>
</tr>
<tr>
<td>• Understands the main functions of the operating system. (DE) (AB)</td>
<td>• Selects, combines and uses internet services. (EV)</td>
<td>• Recognises the audience when designing and creating digital content. (EV)</td>
</tr>
<tr>
<td>• Knows the difference between physical, wireless and mobile networks. (AB)</td>
<td>• Demonstrates responsible use of technologies and online services, and knows a range of ways to report concerns.</td>
<td>• Understands the potential of information technology for collaboration when computers are networked. (GE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Uses criteria to evaluate the quality of solutions, can identify improvements making some refinements to the solution, and future solutions. (EV)</td>
</tr>
</tbody>
</table>
At start of unit: Determine the ‘why’ at the start of the unit of study (Stage 1) as well as the possible topics (the column header names from the Progression Pathways Assessment Framework) that the scheme of work will be covering.

When planning each lesson in a unit of study: Repeat steps 2 - 4 (on the next slide)
CT in the *Planning* part of the cycle

**Step 2: Planning**

- ‘What’ learning outcomes from Assessment Framework (Stage 4)

**Step 2: Planning**

- Step 3: Identify the computational thinking concepts mapped to learning outcomes (Stage 2)

**Step 2: Planning**

- Step 4: Choose activity and consider Classroom Techniques (Stage 3)
Mapping to the framework

**Subject knowledge:**
- Shows an awareness of tasks best completed by humans or computers. (EV)
- Designs solutions by decomposing a problem and creates a sub-solution for each of these parts. (DE) (AL) (AB)
- Recognises that different solutions exist for the same problem. (AL) (AB)

**Classroom technique:**
- Making judgements based on external criteria
- Making trade-offs between competing requirements
- Making judgements about effectiveness (When is good enough good enough?)
- Making comparisons and reaching conclusions
- Interpretation of test results
Example activity: Mapping

**Which classroom techniques?**

**Evaluation**
- Making judgements based on external criteria
- Making trade-offs between competing requirements
- Making judgements about effectiveness (When is good enough good enough?)
- Making comparisons and reaching conclusions
- Interpretation of test results

---

**Human vs. Computer**
Aim: To use what you have learnt from the human robots activity to help you understand which tasks are best completed by a computer and a human.

**Human**
- Printing a magazine
- Cooking a three course meal
- Painting your portrait
- Navigating a maze

**Computer**
- Printing a magazine
- Turning a security light on and off
- Monitoring radioactive substances
- Controlling a garden watering system
- Giving a speech
- Feeding your dog
Spot the CT in the activity
**CT only underpins computer science?**

- Computational thinking underpins the **whole** of the subject of computing

---

**Computational Thinking**

- Uses technology with increasing independence to purposefully organise digital content. (AB)
- Shows an awareness for the quality of digital content collected. (EV)
- Uses a variety of software to manipulate and present digital content: data and information. (AL)
- Shares their experiences of technology in school and beyond the classroom. (GE) (EV)
- Talks about their work and makes improvements to solutions based on feedback received. (EV)
- Collects, organises and presents data and information in digital content. (AB)
- Creates digital content to achieve a given goal through combining software packages and internet services to communicate with a wider audience e.g. blogging. (AL)
- Makes appropriate improvements to solutions based on feedback received, and can comment on the success of a solution. (EV)
- Makes judgements about digital content when evaluating and repurposing it for a given audience. (EV) (GE)
- Recognises the audience when designing and creating digital content. (EV)
- Understands the potential of information technology for collaboration when computers are networked. (GE)
- Uses criteria to evaluate the quality of solutions, can identify improvements making some refinements to the solution, and future solutions. (EV)
Engage all students

The vision **MUST NOT** focus ‘**what**’ you are learning... but ‘**why**’ you are learning it!
In the classroom

*** Producing Skills Outcomes ‘artefacts’ to evidence the Outcomes from Knowledge & Understanding
Does this fit the vision...? No!
Role play: feedback to the teacher on example lessons to draw out computational thinking

Depending on time and the audience:
Choose 2 or 3 of the 5 to roll play
Dinosaurs day out...
Police, Camera and Action!

BREAKING NEWS:
Car chase on streets of French town
Giving those old lessons a face lift!
Scratch calculator
Get with the ‘algor-rhythm’
Binary behind images and secrets

1 Bit - Colour by numbers

2 Bits - Colour by numbers

255 Colours

128 Colours

8 Colours

Lets zoom in!
Binary behind data and the Internet
“Wow… Improves proportion in art and teaches recursion!”

How do these grab you?
* 3.1416
* 6.238673?
* 1.6180399?
Heard of the Golden Ratio?

Can you spot the pattern?
0, 1, 1, 2, 3, 5, 8, 13, 21
Navigating the maze
Influences on planning

• To what extent do you think the following affects how we (as teachers) design our curriculums or CPD events?
  – Our preferred learning styles
  – Our areas of expertise i.e. subject specialism…
  – Our life experiences…

• If you were to teach my curricular or CPD event would they still be creative? would they still contain computational thinking?

• Its about providing a range of activities to meet the different learning styles or approaches to learning
Considerations...

- Other considerations?
  - The students preferred learning styles...
  - The students prior and future learning needs...
  - The students (attendees) areas of expertise i.e. subject specialism i.e RE, PE, Maths...

- Other considerations?
  - The colleagues preferred learning styles...
  - The colleagues prior and future learning needs...
  - The students (attendees) areas of expertise i.e. subject specialism i.e RE, PE, Maths...
First step...

• What are the **challenges** and **opportunities** for creating and delivering a creative computing curriculum underpinned by computational thinking in your school?
Impact in the classroom
Impact in the classroom

• Updated version of OFSTED lesson observation form including prompts for drawing out computational thinking
• Completed example of a lesson observation form focused on computational thinking
Developed by:
Mark Dorling
mark.dorling@computingatschool.org.uk / @MarkDorling
National CPD Coordinator
Computing At School

Concepts and structure were contributed by:
Thomas Ng
CAS Board of Management
School Improvement Adviser (ICT & Assessment)
West Berkshire Council

Computing At School is supported and endorsed by: