Viking Academy Trust

Computing

The VIKING ACADEMY TRUST Computing Policy for 'Ramsgate Arts Primary School' has been written after consultation with staff and following DfE guidance.

Approved by the Trust: Term 1 2018

Reviewed annually: Term 6

Last review date: Term 5 2018

Signed: [illegible]

Chair of Trust

Computing Policy

The Viking Academy Trust

Ramsgate Arts Primary School

Schools in the Viking Academy Trust (VAT)

We start 2017-18 academic year with three schools in the Viking Academy Trust.

These are:

Chilton Primary School
Ramsgate Arts Primary School
Upton Junior School

This Computing Policy is specific to Ramsgate Arts Primary School.

RAMSGATE ARTS PRIMARY SCHOOL COMPUTING EDUCATION POLICY

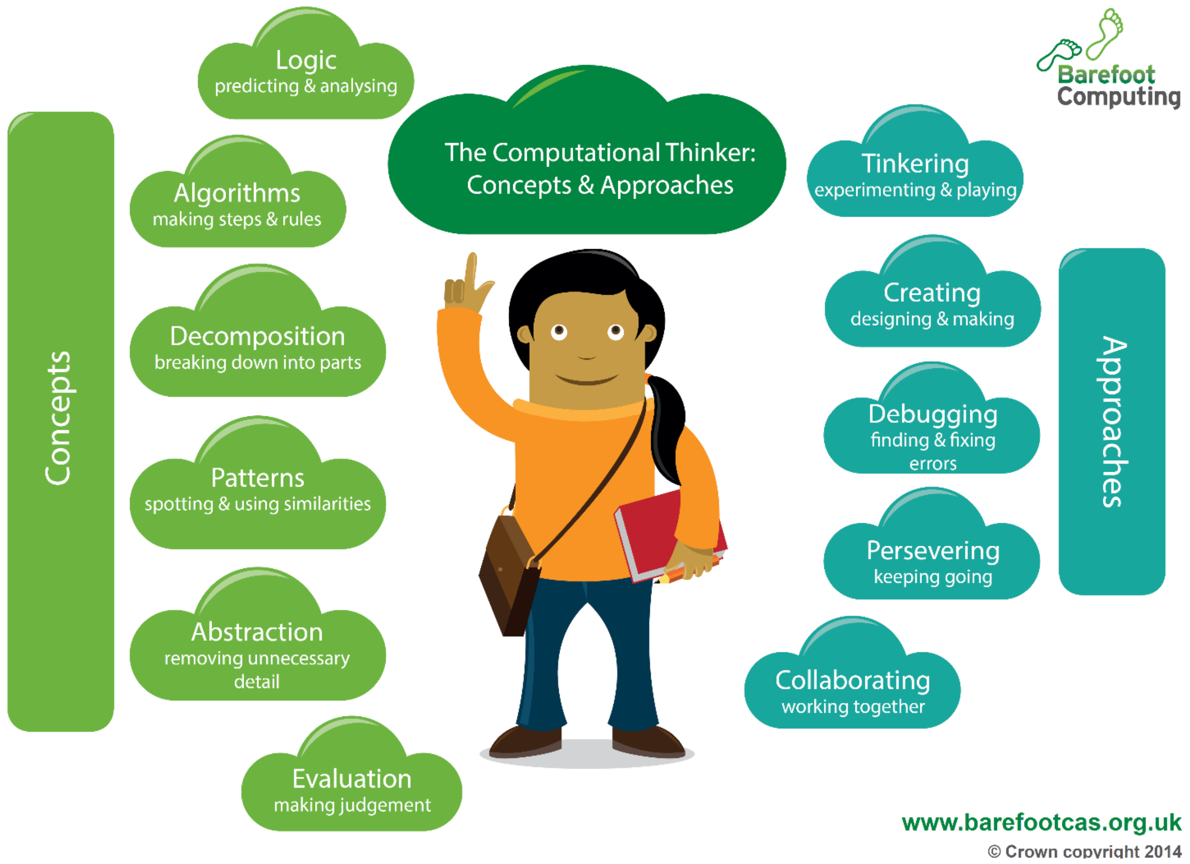
(reviewed September 2017)

What is computational thinking?

"A high quality computing education equips pupils to use computational thinking and creativity to understand and change the world." (Computing National Curriculum)

Computational thinking allows us to develop skills and techniques to help us solve problems effectively, with or without the aid of a computer. Computational thinking is not 'thinking like a computer' – computers are not capable of thought. Rather, it is learning to think in ways which allow us, as humans, to solve problems more effectively and, when appropriate, use computers to help us do so.

Computational thinking involves 6 different concepts and 5 approaches to working:



EYFS

There is lots of opportunity to encourage the building blocks of computational thinking. For example, with support, pupils can work collaboratively to build the highest tower, or to work out the best way to negotiate climbing equipment.

At this level pupils can, with support, understand algorithms, decomposition, and abstraction, for example working out how to 'get dressed for winter' involves decomposing the problem and sequencing instructions (algorithms).

They are also focussing only on the important elements, rather than the detail (abstraction).

KS1

Pupils can sequence simple algorithms using decomposition of simple problems, such as how to grow a plant from seed. They may label the parts of a flower (decomposition), and check with a partner to see if their work is correct (collaborative debugging and evaluation). Pupils can consider whether a problem is best solved with or without a computer.

For example, when finding out about a particular topic, say how methods of transport have changed over time, would it be best to look in a book, or look on the internet? How can they conduct the internet search in the most efficient way?

Similarly, if they wanted to take a picture of something so others could see it, which device, if you have a choice, would be best to use and why?

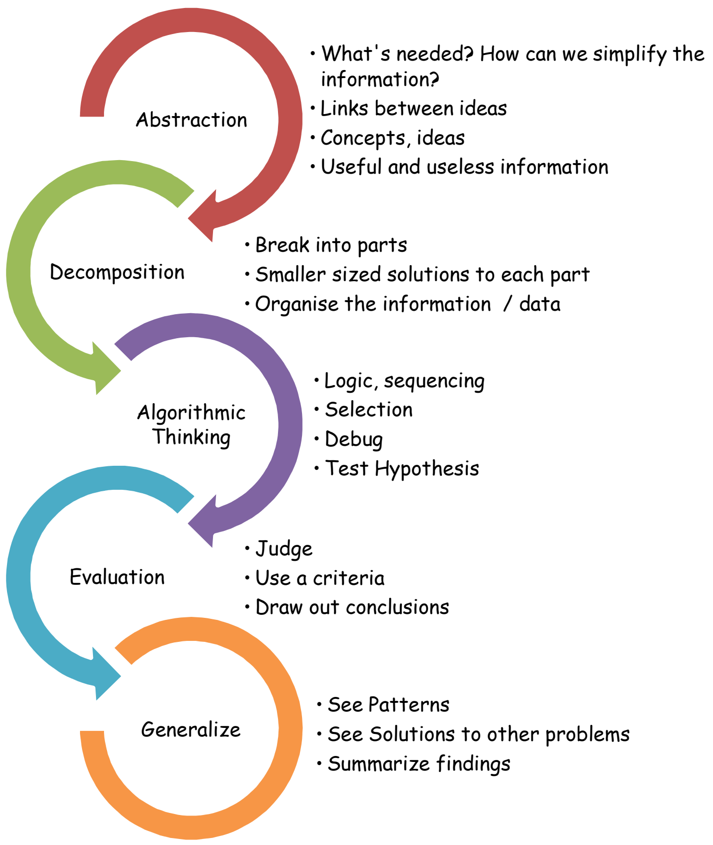
KS2

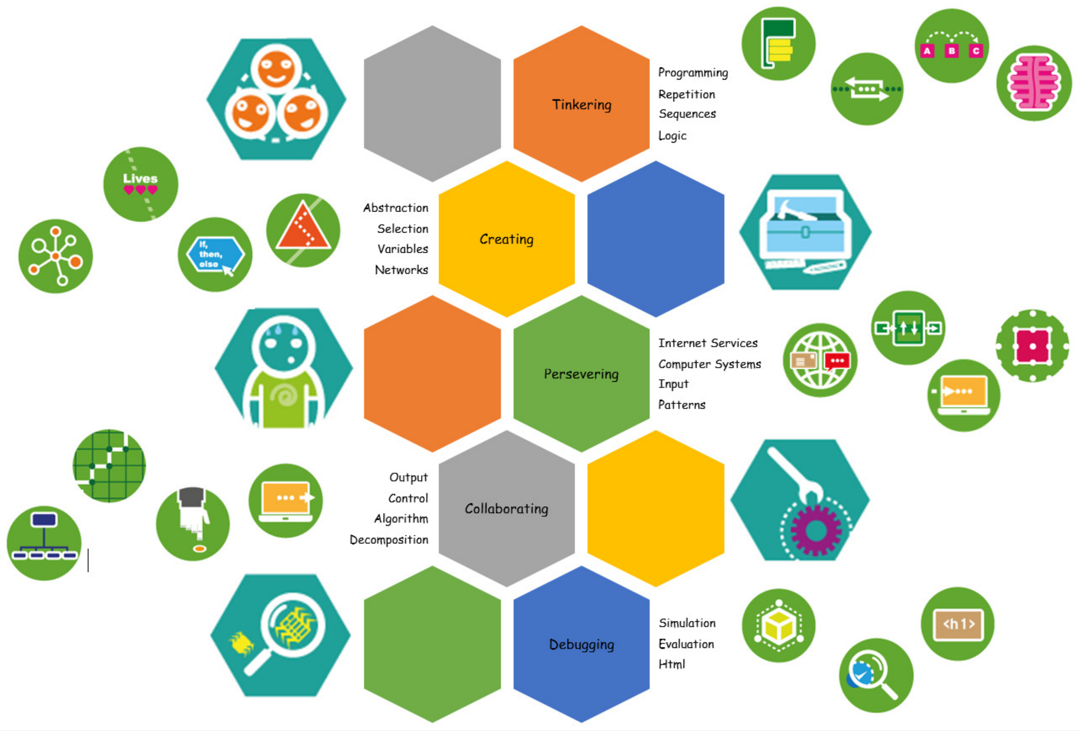
As pupils progress through key stage 2 they can demonstrate increasing levels of computational thinking as their cognitive ability develops; decomposing to an increasing number of levels, designing algorithms and implementing programs with increasing confidence.

They can spot patterns and abstract more readily, focussing on relevant detail only; for example in maths working out that if they complete a multi-step problem in a particular way, are more likely to reach the correct answer (or vice versa!).
Computational thinking approaches become more familiar, for example, pupils can debug a problem more effectively, whether that is finding and correcting grammatical errors in a piece of text or code, either independently or collaboratively.

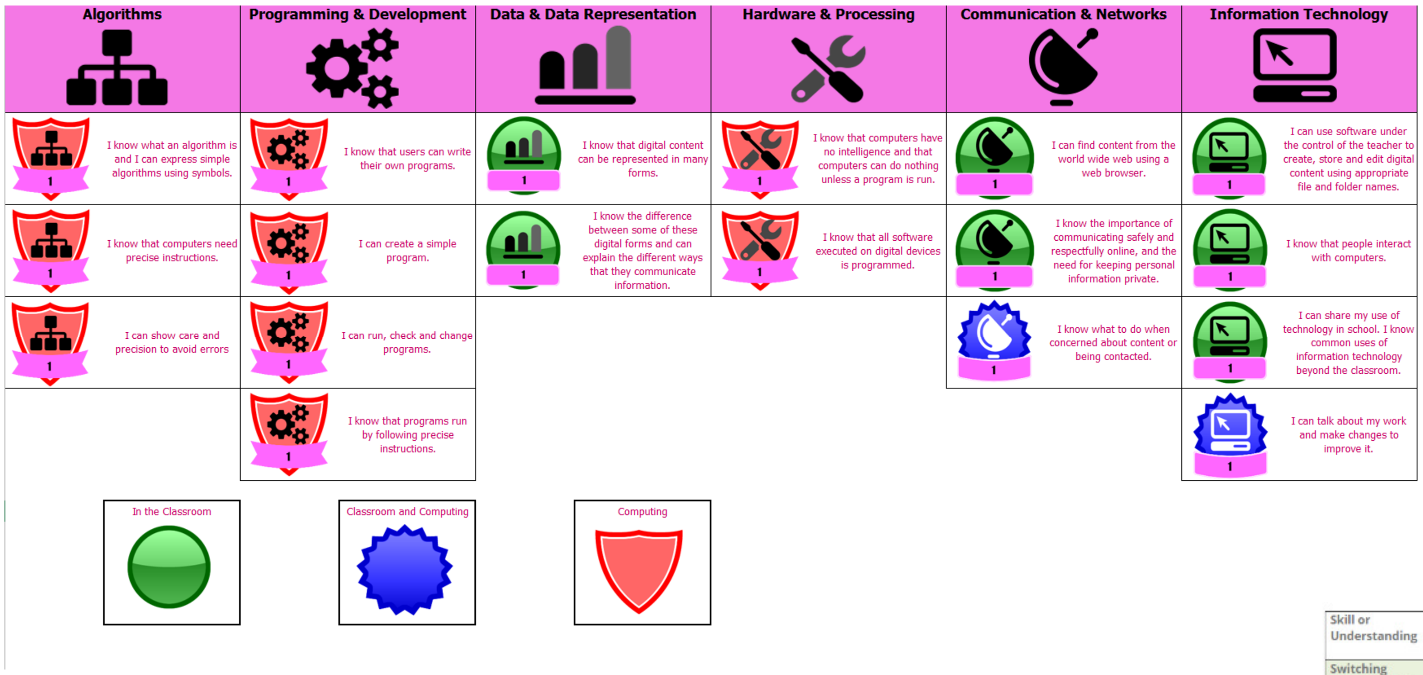
A classroom culture in which collaboration and 'trying-things-out' (tinkering) is actively encouraged, while over-reliance on the teacher is discouraged can help to build pupils' confidence alongside their computational thinking.

(**barefoot**cas.org.uk/wp-content/.../**Computational**-thinking-**Barefoot-Computing**.pdf)
Viking Academy Trust – Understanding the terminology of Computational Thinking


Abstraction
What's needed? How can we simplify the information?
Links between ideas
Concepts, ideas
Useful and useless information
Decomposition
Break into parts
Smaller sized solutions to each part
Organise the information / data
Algorithmic Thinking
Logic, sequencing
Selection
Debug
Test Hypothesis
Evaluation
Judge
Use a criteria
Draw out conclusions
Generalize
See Patterns
See Solutions to other problems
Summarize findings


Tinkering
Programming
Repetition
Sequences
Logic
Creating
Abstraction
Selection
Variables
Networks
Persevering
Internet Services
Computer Systems
Input
Patterns
Collaborating
Output
Control
Algorithm
Decomposition
Debugging
Simulation
Evaluation
Html
Lives
If, then, else
A B C
<h1>

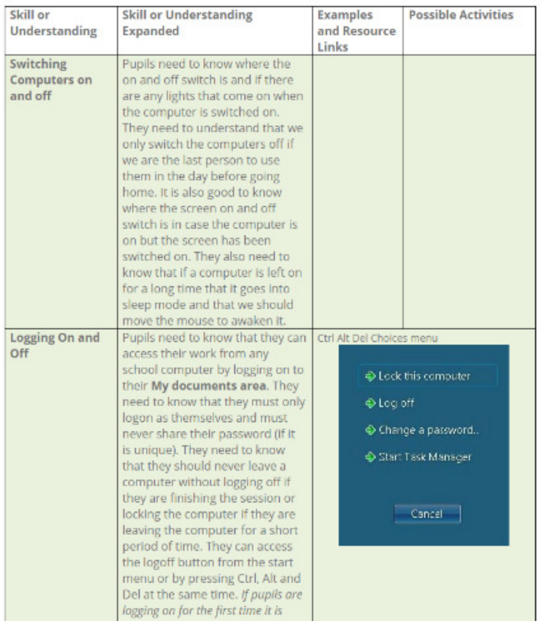
Algorithm	Algorithm (programming) Evaluation	Decomposition	Generalisation	Pattern Spotting	Abstraction	Design	Sequence	Selection	Repetition	Variable Use	Input Output	Debugging
Think of a simple every day algorithm (Remove tooth-paste top, squeeze tooth-paste onto brush)	Identify if an algorithm does what you want it to do read?)	Break a simple everyday algorithm into parts (breakfast, getting changed, walk to school)	Recognise where an idea is adapted and used again as directed by the teacher	Spot simple patterns in code or algorithm (simple regular 2d logo shapes)	Define all the elements in something and then remove the ones that are not needed	Following instructions to create	Follow simple everyday sequences of instructions	use single simple condition	Explore and use simple repetition in music and dance	variable used to hold number or word and reported	simple inputs (keys, mouse click, switch etc) control on or off	Recognise that there is a problem say what problem is
read and follow symbol sequence algorithm (PE Cards, jump, step etc)	Recognise that there are more than one algorithm to do the same thing	Observe a working program and decompose its elements as a class	Pupil chooses to Adapt ideas that they have used to solve similar problems	Spot patterns in algorithm or code and continue the patterns		Add small non critical adaptations	Create simple everyday sequences of instructions	create selection within a loop	Create simple repeat x times loop (Scratch Jr, scratch music notes in repeat loop, repeat x in logo)	multiple non connected variables used	Changing state other than on off such as fast slow bright dim etc	identify where in the code or algorithm bug/ problem occurs
Create simple sequence algorithms using symbols	Recognise that one algorithm may be better	Observe a working program and decompose its elements as an individual				Adapt a given design for a new teacher given purpose	create sequence of symbols	Use single math's operator condition	Create non terminating continuous (forever) loop	variables that change inside a loop	Use sensors to control or report	Debug simple sequence errors independently
Read and follow written sequence algorithms (Forward 3, right 90)	Design an algorithm /code for a specific person or group of people (design a Scratch program for younger pupils)	Create a pro- gram by decom- posing it into parts and solving parts separately				Repurposing ide- as for a pupil chosen purpose	create sequence of simple code that can be easily read	Multiple selection beyond if and else (Scratch use of multiple ifs	loops within loops for a reason	variables inter- acting with other variables		Debug simple repetition, selection & variable errors independently
write simple sequence algorithms using words, use rules algorithms	Evaluate more complex code that does the same thing (Logo design with procedures and with- out, Scratch code with loops and without)						create multiple sequences running concurrently	Multiple conditions using AND OR NOT	Create and use loops that terminates when condition met			Debug repetition, selection & variable errors independently
Read and follow algorithms with selection and repetition							create sequences or multiple sequences where timing is critical (control sprite on					Dividing up code to find where the error is or running Scratch sets of blocks separately
Complete unfinished algorithms with selection and or repetition		**COLOUR KEY**	🟩 MAINLY KS1 ONLY	🟪 KS1 & KS2	🟦 MAINLY KS2 ONLY	🟨 KS2 & KS3						
Create an algo rithm with selec- tion or repetition									cod e-it .co.u k			



Algorithms	Programming & Development	Data & Data Representation	Hardware & Processing	Communication & Networks	Information Technology
I know what an algorithm is and I can express simple algorithms using symbols.	I know that users can write their own programs.	I know that digital content can be represented in many forms.	I know that computers have no intelligence and that computers can do nothing unless a program is run.	I can find content from the world wide web using a web browser.	I can use software under the control of the teacher to create, store and edit digital content using appropriate file and folder names.
I know that computers need precise instructions.	I can create a simple program.	I know the difference between some of these digital forms and can explain the different ways that they communicate information.	I know that all software executed on digital devices is programmed.	I know the importance of communicating safely and respectfully online, and the need for keeping personal information private.	I know that people interact with computers.
I can show care and precision to avoid errors	I can run, check and change programs.			I know what to do when concerned about content or being contacted.	I can share my use of technology in school. I know common uses of information technology beyond the classroom.
	I know that programs run by following precise instructions.				I can talk about my work and make changes to improve it.

Sample Progressions of Assessment and Skills – With Badges.

Pink (Ks1) – Yellow (Ks1/2) – Orange (Ks2) – Blue (Ks2) – Purple (Ks2/3)



Skill or Understanding	Skill or Understanding Expanded	Examples and Resource Links	Possible Activities
Switching Computers on and off	Pupils need to know where the on and off switch is and if there are any lights that come on when the computer is switched on. They need to understand that we only switch the computers off if we are the last person to use them in the day before going home. It is also good to know where the screen on and off switch is in case the computer is on but the screen has been switched on. They also need to know that if a computer is left on for a long time that it goes into sleep mode and that we should move the mouse to awaken it.		
Logging On and Off	Pupils need to know that they can access their work from any school computer by logging on to their **My documents area**. They need to know that they must only logon as themselves and must never share their password (If it is unique). They need to know that they should never leave a computer without logging off if they are finishing the session or locking the computer if they are leaving the computer for a short period of time. They can access the logoff button from the start menu or by pressing Ctrl, Alt and Del at the same time. *If pupils are logging on for the first time it is*	Ctrl Alt Del Choices menu	

Sample Skills for the "Class-teacher" and used to support Digital Literacy

EYFS

The use of IPads within the classroom (APPS, Cameras, a tool)
Control technologies – Beebots and Bluebots and remote control cars etc
Devices to record sounds, create videos and draw artwork
Use the Interactive Whiteboard to collaborate with others (take turns and share ideas)

KS1

The use of IPads within the classroom (APPS for control, develop debugging skills, logical reasoning and a learning tool)
Control technologies – Beebots and Bluebots and remote control cars etc
Devices to record sounds, create videos and draw artwork
Use the Interactive Whiteboard to collaborate with others (take turns and share ideas)
Support learning in English, Maths and other curriculum areas

Ks2 (Y3/4)

The use of IPads within the classroom (APPS, Cameras, a tool). Support Computing
Chromebooks / Windows Tablets to support online research, flash based resources, VLE. Support the Computing Curriculum with creating algorithms, debugging, creating and solving problems and puzzles.
Control technologies – SPheros SPRK – Probots
Devices to record sounds, create videos and draw artwork
Use the Interactive Whiteboard to collaborate with others (take turns and share ideas)

Ks2 (Y5/6)

Chromebooks to support online research, flash based resources, VLE. Support the Computing Curriculum with creating algorithms, debugging, creating and solving problems and puzzles.
Windows tablets to prepare children for the future at home and at work. Minecraft and Kudo for computing
Mbot – build and control own robots (Bluetooth)

Devices to record sounds, create videos and draw artwork

Collaborate with ideas, concepts and use technologies to support.

Aims

- To enable children to become autonomous, independent users of computing, gaining confidence and enjoyment from their activities
- To develop a whole school approach to computing ensuring continuity and progression in all strands of the computing National Curriculum
- To use computing as a tool to support teaching, learning and management across all areas of the curriculum
- To provide children with opportunities to develop their computing capabilities in all areas specified by the Curriculum.
- To ensure ICT is used, when appropriate, to improve access to learning for pupils with a diverse range of individual needs, including those with SEN and disabilities
- To maximise the use of computing in developing and maintaining links between other schools, the local community including parents and other agencies.

Objectives

In order to fulfil the above aims it is necessary for us to ensure:

- a continuity of experience throughout the school both within and among year groups
- the systematic progression through key stages 1 & 2
- that the National Curriculum programmes of study and their associated strands, level descriptions and attainment target are given appropriate coverage
- that all children have access to a range of ICT resources
- that computing experiences are focussed to enhance learning
- that cross curricular links are exploited where appropriate
- that children's experiences are monitored and evaluated
- that resources are used to their full extent
- that resources and equipment are kept up to date as much as possible
- that staff skills and knowledge are kept up to date

By the end of Key Stage 1 pupils should be taught to:

- Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following a sequence of instructions
- Write and test simple programs
- Use logical reasoning to predict and computing the behaviour of simple programs
- Organise, store, manipulate and retrieve data in a range of digital formats
- Communicate safely and respectfully online, keeping personal information private, and
- recognise common uses of information technology beyond school.

By the end of key stage 2 pupils should be taught to:

- Design and write programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- Use sequence, selection, and repetition in programs; work with variables and various forms of input and output; generate appropriate inputs and predicted outputs to test programs.
- Use logical reasoning to explain how a simple algorithm works and to detect and correct errors in algorithms and programs.
- Understand computer networks including the internet; how they can provide multiple services, such as the world-wide web; and the opportunities they offer for communication and collaboration
- Describe how internet search engines find and store data; use search engines effectively; be discerning in evaluating digital content; respect individuals and intellectual property; use technology responsibly, securely and safely.
- Select, use and combine a variety of software, (including internet services), on a range of digital devices to accomplish given goals, including collecting, analysing, evaluating and presenting data and information.