Skills and Knowledge Progression – Computing 2023

This progression is based on documents by Phil Bagge from the Computing Hias team. The HIAS team split information into three strands:

Declarative knowledge - Static facts or knowledge stored in your memory (**what**) = KNOWLEDGE
Procedural knowledge - How to perform a specific skill or task (**how**) = SKILLS
Conditional Knowledge - When to use declarative and procedural knowledge (**when & why**) = SKILLS (APPLICATION)

NB: detailed breakdowns of skills can be found in the folder entitled 'Detailed breakdowns of skills in computing'.

KS1 NC, Skills and Knowledge

	Computer Science	Information Technology/Digital Literacy	E-safety
KS1 National Curriculum	**Pupils should be taught to:** - understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. - create and debug simple programs. - use logical reasoning to predict the behaviour of simple programs.	**Pupils should be taught to:** - use technology purposefully to create, organise, store, manipulate and retrieve digital content. - recognise common uses of information technology beyond school.	**Pupils should be taught to:** - use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.
KS1 Knowledge – I…	Know that an algorithm is a set of precise instructions that can be understood by another human. Know that a digital device contains coded instructions to make it do things. Know that digital devices can come in many shapes and sizes. Know that computers are one form of digital device. Know that an algorithm can be used to plan a program on a digital device.	Know I can create lots of different types of things on digital devices. Know that information technology is used outside school. Input data into digital devices. Know that keyboards can be digital or onscreen. Know that highlighting is used before making other formatting changes.	Know that some information is private (only for me and trusted people) and some information is public (available for everyone). Know I can use online technologies to find information (web browser, Smart devices like Alexa or Siri). Know I will come across things online that I like and things I don't like. Know that some things online are for adults, not me. Know I should tell a trusted adult if I see things online that make me feel sad, upset or confused. Know I should only share personal information after checking it first with a trustworthy person.
KS1 Skills – I can…	Plan an algorithm that can be turned into code on a digital device. Debug code that does not do what I want it to do. Look at code to work out what a program will do when the code is run.	Use a keyboard to input letters and symbols. Use a pointing device to select and change items. Use a digital device to word process text. Use a digital device to present my ideas. Use a digital device to draw pictures. Use a digital device to take and change photos. Use a digital device to take and change videos. Use a digital device to record sounds. Save my digital creations by naming them after what they are about. Create a folder and name it after what it will store. Find my saved documents and open them. Identify where information technology is used in my house. Identify where some information technology is used in my neighbourhood. Change text including the size, font, colour, bold, italicised.	Identify personal information about me such as my name, age, birthday, where I live and go to school. Decide who is trustworthy based on their relationship to me or their jobs.

KS2 NC, Skills and Knowledge			
	Computer Science	**Information Technology/Digital Literacy**	**E-safety**
KS2 National Curriculum	**Pupils should be taught to:** - design, write and debug 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 - use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.	**Pupils should be taught to:** - 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. - use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content. - select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.	**Pupils should be taught to:** - use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

KS2 Knowledge – I…	Know that an algorithm is a set of precise instructions that can be understood by another human. Know that an algorithm can be used to plan a program on a digital device. Know that a digital device contains coded instructions to make it do things. Know that digital devices can come in many shapes and sizes. Know that computers are one form of digital device. Know that inputs put-in information into digital devices. Identify common input devices like buttons, dials, mouse, keyboards, microphones, cameras. Know that outputs put-out information from digital devices. Identify common output devices like screens, speakers and printers.	<u>Publishing</u> Know that documents can be created that combine text and graphical elements (shapes, pictures, tables) for many purposes (adverts, posters, fliers etc). Know that desktop publishing documents were originally designed to be printed but now are more likely to be published on the web. <u>Green Screen</u> Know that green screen technology works by identifying a colour to remove from a video or picture and then filling that colour space with another picture or video. <u>Presentation</u> Know that the presentation Media is used when information needs to be shared by one person to many people. Know that the presenter can be present or absent. Know that if the presenter is present not all information should be shared on the slides. Know that if the presenter is absent then all information needs to be present or with the slide as sound or video inlay. <u>Spreadsheets</u> Know that a spreadsheet is used to organise, store and analyse data. Know that spreadsheets can work with numbers, text or graphics, but has greater functionality with numbers. <u>Stop Motion</u> Know that stop motion involves viewing multiple pictures that are slightly different, quickly. <u>Google Forms</u> Know a single set of questions can be shared with many people through the web, allowing each person to input their answers separately. Know that each person's answers to a survey can be viewed independently or combined with other people's answers. <u>Web Research</u> Know that first results on a search are those that have been linked to by the largest number of other websites. Understand that data comes in different formats, text, images and videos. Understand that data comes in different sizes; text is the smallest and videos are the largest. <u>Web Building</u> Know that websites are documents hosted on computers, that could be anywhere in the world. Know that websites can be accessed publicly or shared with a smaller group. <u>Word Processing</u> Know that highlighting is used before making other formatting changes.	<u>Web Building</u> Know the positives and negatives of working with an online group.

KS2 **Skills – I can…**	Plan algorithms that can be turned into code on a digital device. Debug code that does not do what I want it to do. Break a large programming task up into smaller parts and solve these parts separately. Program using simple sequence. Program using repetition. Use the Scratch programming environment. Program using conditional selection. Program using variables. Program using procedures. Design and program a physical system.	<u>Publishing</u> Work with graphics, shapes, multiple objects and tables. Change the view of the document. <u>Green Screens</u> Create, edit and export a green screen project. <u>Presentation</u> Create a new slide and add things, e.g. graphics, transitions, animations, videos and sound, to create a desired effect. <u>Spreadsheets</u> Create a spreadsheet, enter data and use the graphing and formula functions. <u>Stop Motion</u> Create, edit and improve a stop motion animation. <u>Surveying</u> Set up a survey with different question types. Complete a survey. Analyse the responses to surveys, including using graphs. <u>Web Research</u> Choose an appropriate search engine. Use a search engine to find information and answer questions. Select, copy and paste information from a website. <u>Web Building</u> Create, edit and improve a website. Use graphics, photos and other images to create desired effects. <u>Word processing</u> Change text including the size, font, colour, bold, italicised to create desired effects. Manipulate the layout of items on a page to create desired effects.	<u>Green screens</u> Keep myself safe when using green screen technology. <u>Web research/Web building</u> Analyse websites to know if they are trustworthy. Make sensible decisions when using the internet.

Appendix 1 – KS1 detailed knowledge planning objectives

Detailed Keyboard Knowledge

I can find common letters quickly on a QWERTY keyboard.
I can use the space key to insert gaps between words.
I can use return or enter to move text onto a new line.
I can use arrow keys to move the cursor around within text to aid accuracy in highlighting and deleting.
I can use shift for capitals lowercase text avoiding caps locks unless lots of text needs to be in capitals.
I can use shift for other symbols available on the top of the key or to switch between upper and lower case.
I can use the delete key to delete right.
I can use backspace to delete left.

Detailed Drawing Knowledge

I can use a pen or brush tool to draw with the mouse or trackpad.
I can change the colour of my pen or paint brush.
I can use a digital rubber to remove markings.

Detailed Pointing Device Knowledge

I can move the cursor to the right place on the screen using mouse or trackpad lifting mouse if necessary.
I know where right and left buttons are on my trackpad (if using laptops)
I can hold the mouse still while clicking buttons
I can single left click to open links on a browser
I can double left click to open software
I can left click and hold to highlight text
I can double left click to highlight one word

Detailed Presentation Knowledge

I can create a new slide.
I can choose an appropriate format for a new slide that matches the information I need to share, text only, text and graphics.
I can add text to a new slide whilst keeping text to a minimum to avoid overload.
I can import a graphic into a slide that is appropriate for the topic .
I can change between building and presenting modes and know when both should be used.

Detailed Saving Knowledge

I can open a saved document within a program (windows).
I can open a saved document outside a program (windows).
I can save a document into a previously created folder (windows).

Detailed Word Processing Knowledge

I can highlight a line of text by left clicking on left of line.
I can highlight a single work by double left clicking on it.
I can highlight by left clicking and holding down whilst dragging over text.
I can change text size for titles or headings.
I can change text font using no more than two fonts in any document one for headings one for main body.
I can change text to bold for titles, headings or emphasis.
I can change text to italics for emphasis or quotes.
I can change text colour for emphasis.
I can highlight text for document review only.
I can use bullet points for lists where the order is not important.
I can use numbered points for lists where the order is important.
I can align text left and use this for most documents.
I can align text right for emphasis or to line text up with other elements.
I can align text in the centre for titles.

Appendix 2 – KS2 detailed <u>programming/computer science knowledge</u> planning objectives

<u>Detailed Debugging Knowledge</u>
I can run sections of code separately to try and identify where the bug is.
I know that some bugs are just unfinished code rather than anything wrong with my existing code.
I can read my code out loud to help identify bugs.
I can explain my code to a partner to help me identify bugs.
<u>Detailed Sequence Knowledge</u>
I know that some blocks are timed and some blocks are not making them run very quickly.
I know that the order of most sequences is very important but for a few sequences this is not true.
<u>Detailed Repetition Knowledge</u>
A loop is a sequence of instructions that are repeated.
Indefinite means that the number of loops completed will not be known by the programmer.
Definite means the number of loops are known.
A **count-controlled-loop** ends after the count is finished and is a definite loop.
An **infinite** loop never ends and is an indefinite loop.
A **condition-ends-loop** is ended by a condition and is an indefinite loop.
<u>Repetition Algorithmic Knowledge</u>
I can write a loop as an algorithm using appropriate loop language such as do so many times, loop always, loop until or repeat x times.
I can indent actions inside a loop to make it clear they are inside the loop.
<u>Repetition Programming Knowledge</u>
I can use a count-controlled-loop where I want an sequence of actions to take place for a specific amount of time or over a specific distance.
I can use an indefinite infinite loop where I don't want the loop to end once it has started.
I can use an indefinite loop that is ended by a condition when I need the user or another. program mechanism to interact with the loop end
I can nest count-controlled-loops where I want to take advantage of the cumulative effect of inner loops.

<u>Detailed conditional selection knowledge</u>
Conditions can be met (true) or not met (false) there is no middle option.
A true of false condition branches outcomes into two paths.
A condition is only checked when it is reached in the flow of control.
Conditions can be combined using AND or OR (KS3+).
Conditions can be inverted using NOT (KS3+).
<u>Conditional Selection Algorithmic Knowledge</u>
I can indent actions that result from a condition being met or not met to separate them from actions not affected by selection.
I can use if to indicate that a condition will follow in my algorithm.
I can draw a flow of control diagram to show how a condition branches and what actions are on each branch.
<u>Conditional Selection Programming Knowledge</u>
I can use a **condition-starts-action** selection block to run code where a condition is true.
I can use a **condition-switches-between-actions** selection block to run code where a condition is true or false.
I can combine multiple **condition-starts-action** selection blocks to create multiple branches.
I can use a selection block within a forever loop to check a condition over and over again.

<u>Detailed Variable Knowledge</u>
Variables are used to store information to be referred to and changed in a computer programme or algorithm.
Variable have a name and a value.
Read the variable name but act on the variable value.
Variables can be global, affect the whole programme or local only affecting a part.
When a value is attached to a variable we call it assigning.

<u>Variable Algorithmic Knowledge</u>
I can name a variable after the task it performs in my algorithm plan.
I can name a variable using one word and camelCase or under_score to avoid confusion later in text based programming languages.
I can avoid naming a variable using names used for other variables, procedures or lists to avoid confusion later in text based programming languages.
I can assign a value to a variable at the start of my algorithm plan so that the algorithm always works in the same way every time (initialisation).
I can add or subtract value to a variable where the variable is a number.
<u>Variable Programming Knowledge</u>
I can create a variable and show the value on the screen in Scratch so either I or the user can see the value.
I can use a variable as a placeholder to refer to a value input by a user.
I can use a variable to interact with other blocks using join blocks where needed.
<u>Detailed Simple Procedure Knowledge</u>
A procedure is a small named section of a program that performs a specific task.
A procedure should be named uniquely.
A procedure is called or run by its name.
A procedure can be run many times in a programme.
<u>Procedure Algorithmic Knowledge</u>
I can name a procedure after the task that the instructions carry out.
I can name a procedure using camelCase or under_score to avoid problems creating procedures in text based programming languages.
I can name a procedure uniquely to avoid problems with variables or lists in text based .programming languages
<u>Procedure Programming Knowledge</u>
I can create a procedure in scratch from My Blocks
I can create a main program and procedures to call the procedures from within the main programme.
I can look for processes I need to use many times and create them as procedures to save time

Appendix 3 – KS1 and KS2 detailed <u>Information Technology and Digital Literacy</u> support documents (from the HIAS computing team)

These plans supplement the above skills and knowledge with more detail, as well as including:

- *prior knowledge*
- *icon examples to use when modelling*
- *keywords etc*
- *Assessment criteria*

These plans won't replace a STP – they aren't broken down into a number of sessions, they have no context etc - but can be used as a starting point. Suggested year groups are noted also below.

Desktop Publishing – Y4, Y5, Y6
Green Screens – KS2
Presentation Media – Y3, Y4, Y5
Spreadsheets – Y5, Y6
Stopmotion Animation – Y3, Y4, Y5, Y6
Surveying – Y4, Y5, Y6
Web Research – Y2, Y3, Y4, Y5, Y6
Collaborative Web Building – Y4, Y5, Y6
Word Processing – Y1, Y2, Y3, K4
Keyboard Knowledge – Y1, Y2