

Teaching Guide.

Starlink: Battle for Atlas

While the curriculum material within this workshop is suitable for all ages, please be aware that the video game used in this context is rated PEGI 7 - suitable for ages 7+ only. Should you choose to use this workshop with a younger audience, we advise you to use a different game that carries a PEGI 3 rating. For more information about PEGI ratings please visit AskAboutGames <https://www.askaboutgames.com/> or the Video Standards Council Rating Board <https://videostandards.org.uk/RatingBoard/>.

Introduction

This resource was originally developed by Magic Makers which is a Paris-based coding school developing creative programming workshop for kids. Since 2014, they have introduced more than 10 000 kids to the basics of coding, with the ultimate objective of contributing to the development of creative programming in France. In preparation of the launch of Starlink: Battle for Atlas on October 16 2018, Ubisoft and Magic Makers collaborated to create a dedicated coding program based on the universe of the game.

Digital Schoolhouse has further developed these resources adding unplugged activities and versions of the tutorials to allow the use of Scratch, Construct 3 or Unity to develop the Starlink game. All of the workshop is then embedded into their educational philosophy and framework, including the use of PRIMM for teaching programming skills. This workshop begins by demonstrating each of the 3 programming constructs using party games and then moves into demonstrating them within each of the programming languages / platforms available.

Contents

Learning Outcomes	3
Files/Resources.....	4
Session Overview.....	7
Digital Schoolhouse Progression Matrix.....	21
Algorithms	21
Programming	22
Data	23
Hardware and Software.....	24
Communication	25
IT.....	26
Computing Programmes of Study Links.....	27
Computational Thinking Strands	29

Learning Outcomes

- 1 To be able to understand what is meant by the programming construct sequence
- 2 To be able to understand what is meant by the programming construct selection
- 3 To be able to understand what is meant by the programming construct iteration
- 4 To be able to identify sequence, selection and iteration within a program
- 5 To be able to use sequence, selection and iteration within their program
- 6 To be able to dry run / test their program to see if it works and achieves desired results
- 7 To be able to debug their program
- 8 To be able to evaluate the effectiveness of their solution

Files/Resources

U = Unplugged activity, B = Beginner activity, I = Intermediate activity, A = Advanced activity

Filename	Resource Type	Purpose/Description	Activity No
DSH-tutorial-starlink - HTML5 (Published)	HTML5	All workshop content inc. assessment and PRIMM	All
DSH-teaching-presentation-starlink-unplugged	PowerPoint	Standalone teaching presentation for unplugged elements of the Starlink workshop	U1 – U5
STARLINK_TRAILER_E3 2018 GEMINI REVEAL TRAILER_11-06_2.30pm LA time_UK	Video	Trailer for Starlink	Intro
STARLINK BATTLE FOR ATLAS - CHRIS HADFIELD ON ESSENTIAL ASTRONAUT SKILLS	Video	Chris Hadfield explains the essential skills for becoming an astronaut	Intro
DSH-party-game-flowcharts-starlink.pdf	Activity sheet	Copies of the party game flowcharts	U1 – U5
Scrap paper	Activity resource	Consequences	U1
Pens / Pencils	Activity resource	Consequences	U1
Parcel	Activity resource	Pass the Parcel A prize wrapped in multiple layers of news paper or magazine pages, between each layer alternate a sweet or small prize and a dare written on a slip of paper	U4
Music	Activity resource	Pass the Parcel Background music to play while the parcel is being passed	U4
6 sided standard dice	Activity resource	Chocolate game	U5
Wrapped large chocolate bar (e.g. Dairy Milk)	Activity resource	Chocolate game	U5

Hat	Activity resource	Chocolate game	U5
Gloves	Activity resource	Chocolate game	U5
Scarf	Activity resource	Chocolate game	U5
Plate	Activity resource	Chocolate game	U5
Knife	Activity resource	Chocolate game	U5
Fork	Activity resource	Chocolate game	U5
Scratch 3 (https://scratch.mit.edu/)	Website	Beginner programming activities	B1 – B5
Construct 3 (https://editor.construct.net/)	Website	Intermediate programming activities	I1 – I5
Unity (https://unity.com/learn/education)	Software	Advanced programming activities	A1 - A5
Visual Studio for Unity (https://visualstudio.microsoft.com/vs/unity-tools/)	Software	Advanced programming activities	A1 - A5
DSH-tutorial-starlink-SCORM	SCORM	Step by step instructions for Scratch 3, Construct 3 and Unity inc. assessment and PRIMM (no unplugged activities)	B1 – B5, I1 – I5 and A1 – A5
DSH-teaching-presentation-starlink-Scratch	PowerPoint	Standalone step by step instructions for scratch	B1 – B5
DSH-teaching-presentation-starlink-construct	PowerPoint	Standalone step by step instructions for Construct 3	I1 – I5
DSH-teaching-presentation-starlink-Unity	PowerPoint	Standalone step by step instructions for Unity	A1 – A5
Explosion	PNG image file	Image asset for Starlink game	B1 – B5, I1 – I5 and A1 – A5
Meteor	PNG image file	Image asset for Starlink game	B1 – B5, I1 – I5 and A1 – A5

Ship	PNG image file	Image asset for Starlink game	B1 – B5, I1 – I5 and A1 – A5
Sonatus	PNG image file	Image asset for Starlink game	B1 – B5, I1 – I5 and A1 – A5
Starfox	PNG image file	Image asset for Starlink game	B1 – B5, I1 – I5 and A1 – A5
Primed for Battle	MP3 sound file	Sound asset for Starlink game	B1 – B5, I1 – I5 and A1 – A5
Unity_001_Introduction*	MP4 video file	Step-by-step instructional video for creating game in Unity	A1 – A5
Unity_002_ResourcesAndBackground*	MP4 video file	Step-by-step instructional video for creating game in Unity	A1 – A5
Unity_003_PlayerMovement*	MP4 video file	Step-by-step instructional video for creating game in Unity	A1 – A5
Unity_004_Meteors*	MP4 video file	Step-by-step instructional video for creating game in Unity	A1 – A5
Unity_005_ExplosionEffect*	MP4 video file	Step-by-step instructional video for creating game in Unity	A1 – A5
Unity_006_Lasers*	MP4 video file	Step-by-step instructional video for creating game in Unity	A1 – A5
How to add a score counter into your Unity 2D game_ Easy*	MP4 video file	Step-by-step instructional video for adding scores to a game in Unity	A5
Video_ExportGame*	MP4 video file	Step-by-step instructional video for exporting game from Unity	A1 – A5

* These video files are contained in the HTML5 and SCORM packages and the step-by-step Unity ppt, subtitled versions are also available

PLEASE NOTE: The activities outlined in this workshop pack are a suggested outline of how the workshop can be delivered. It is envisaged that teachers will adapt the resources and the organisation of them according to the needs of their class.

Session Overview

If you are using the HTML5 version of these resources, you will find all content in a single online resource. If you intend to use the SCORM version of the resources via your VLE, it is intended that SESSION 1 should be delivered using the unplugged teaching presentation before students begin the programming tutorials in the SCORM package. Below you will find references to the separate resources available should you want to deliver the programming tutorials as stand-alone sessions.

U = Unplugged activity, B = Beginner activity, I = Intermediate activity, A = Advanced activity

SESSION 1

Activity No	Session Content / Activity	Resources Used
	Welcome and Introduction	DSH-teaching-presentation-starlink-unplugged.ppt – slides 1 - 2 or Starlink HTML5 - slides 1 - 2
	Ask students what they think Starlink is, some may be aware of the game, others not. Once you have had a discussion then show students the video advertising Starlink.	DSH-teaching-presentation-starlink-unplugged.ppt – slides 3 - 8 or Starlink HTML5 - slides 3 - 8
	Ask students what they think Starlink has to do with Computer Science, after a short discussion then show students the video of astronaut Chris Hadfield explaining what the essential skills for living in space are – read the slide explaining that we will be learning how to program the machines	STARLINK_TRAILER_E3 2018 GEMINI REVEAL TRAILER_11-06_2.30pm LA time_UK STARLINK BATTLE FOR ATLAS - CHRIS HADFIELD ON ESSENTIAL ASTRONAUT SKILLS

	Introduce the 3 programming constructs – explain that these are the basic ‘building blocks’ of any programming language	DSH-teaching-presentation-starlink-unplugged.ppt – slide 9 or Starlink HTML5 – slide 9
	Explain what ‘sequence’ is and how it is used in a programming language	DSH-teaching-presentation-starlink-unplugged.ppt – slide 10 or Starlink HTML5 – slide 10
U1	Play a game of consequences: Sit students in a circle and hand out a piece of paper and a pen / pencil to each student Explain that students should make up an answer to the question asked by the teacher, write this down at the top of their piece of paper and then fold over the top of the paper to hide their answer. The paper should then be passed on to the students right and the process repeated on the piece of paper the student has just been given, this time with the next answer being written below the folded over top so that all the answers are on the same side of the paper when the paper is unfolded at the end of the game. The following questions should be asked in this order: What is the name of a female celebrity? What is the name of a male celebrity? What is the name of a place where people might meet? What might someone say to someone else? What might their reply be? What is the consequence?	DSH-teaching-presentation-starlink-unplugged.ppt – slides 11 - 13 or Starlink HTML5 - slides 11 – 13 DSH-party-game-flowcharts-starlink.pdf Scrap paper Pens / Pencils

	Once all the questions have been answered ask students to pass the paper on one last time and then open out the sheet that has been handed to them. Collect in the sheets and then read the following story, adding in the words and phrases from one of the sheets you have just collected in (or you can ask the students to read their own but by collecting them in you can censor anything inappropriate) Once upon a time female celebrity met male celebrity at the location, she said phrase 1 and he replied phrase 2. The consequence was the consequence	
	Explain that consequences is an example of a game that uses sequence, the instructions to play the game must be followed in order otherwise the game does not work. Discuss how this is how a computer follows programs when they are written in sequence	DSH-teaching-presentation-starlink-unplugged.ppt – slides 11 - 13 or Starlink HTML5 - slides 11 – 13 DSH-party-game-flowcharts-starlink.pdf
	Explain what ‘selection’ is and how it is used in a programming language	DSH-teaching-presentation-starlink-unplugged.ppt – slide 14 or Starlink HTML5 - slide 14
U2	Play a game of Simon Says!: Explain to students that they should only do the action that the teacher says if they say Simon says first, e.g. ‘Put your hands on your head’ should not be followed but ‘Simon says put your hands on your head’ should be. If any students do the action when Simon hasn’t said they are out.	DSH-teaching-presentation-starlink-unplugged.ppt – slides 15 - 17 or Starlink HTML5 - slides 15 – 17

	Play a few rounds of the game	DSH-party-game-flowcharts-starlink.pdf
	Explain that Simon Says! Is an example of a game that uses selection as the instruction should only be carried out if it is preceded with the words 'Simon says'. Discuss how this is how a computer follows programs written using selection	DSH-teaching-presentation-starlink-unplugged.ppt – slides 15 - 17 or Starlink HTML5 - slides 15 – 17 DSH-party-game-flowcharts-starlink.pdf
	Explain what 'iteration' is and how it is used in a programming language	DSH-teaching-presentation-starlink-unplugged.ppt – slide 18 or Starlink HTML5 – slide 18
U3	Play a game of Hide and Seek: Choose one student to be the seeker, they should stand facing the wall with their eyes closed and count to 100 (you can shorten this if the students take less time to hide). While the seeker counts, the rest of the students should hide. They should then stay in their hiding place until either they are found by the seeker or the teacher tells them that they are the last to be found and are therefore the winner!	DSH-teaching-presentation-starlink-unplugged.ppt – slides 19 - 21 or Starlink HTML5 - slides 19 - 21 DSH-party-game-flowcharts-starlink.pdf
	Explain that Hide and Seek is an example of a game that uses iteration (or a loop) because you hide until you are found. Discuss that this is how a computer follows programs that use an until loop. Also discuss that there are different types of loop that work in slightly different ways	DSH-teaching-presentation-starlink-unplugged.ppt – slides 19 - 21 or Starlink HTML5 - slides 19 - 21

		DSH-party-game-flowcharts-starlink.pdf
	Explain that multiple constructs can be used together in the same program	DSH-teaching-presentation-starlink-unplugged.ppt – slide 22 or Starlink HTML5 - slide 22
U4	Play a game of Pass the Parcel: Prepare a parcel in the way outlined in the Files/Resources section. Students should sit in a circle with the wrapped parcel in the centre. Play some background music, students should pass the parcel to their right all the time the music is playing. As soon as the music stops, the person holding the parcel at that moment can remove 1 outer layer. If they find a sweet or small prize they can keep it but if they find a dare they have to do it. Once the layer has been removed and the dare/ prize found and completed, the music should be restarted, and the parcel passed on again. Repeat the previous steps until the bigger prize in the centre is unwrapped. Once this happens it is the end of the game and the student that has found the big prize gets to keep it	DSH-teaching-presentation-starlink-unplugged.ppt – slides 23 - 25 or Starlink HTML5 - slides 23 - 25 DSH-party-game-flowcharts-starlink.pdf Parcel prepared as described in the Files/Resources section Music
	Explain that Pass the Parcel is an example of a game that uses multiple programming constructs, iteration because the parcel is passed while the music is playing and while there are still layers to be removed. It uses sequence because every time the music stops a layer is removed and selection because after the layer is removed either a prize is kept, or a dare is completed dependant on what is found inside. Discuss how this is similar to using multiple programming constructs when writing a program	DSH-teaching-presentation-starlink-unplugged.ppt – slides 23 - 25 or Starlink HTML5 - slides 23 - 25 DSH-party-game-flowcharts-starlink.pdf
	OPTIONAL – you may only want to do the following activities for classes that are doing either INTERMEDIATE or ADVANCED mode	

	Explain what 'nesting' is and how it is used in a programming language	DSH-teaching-presentation-starlink-unplugged.ppt – slide 26 Or Starlink HTML5 – slide 26
U5	Play a game of the Chocolate Game: Students should sit in a circle, in the middle of the circle should be a hat, scarf, gloves and a wrapped chocolate bar on a plate with a knife and fork. Students should take it in turns to roll a dice. If a 6 is rolled they get to go into the middle of the circle, put on the hat, gloves and scarf and attempt to open and then eat the chocolate bar using the knife and fork provided. The dice should continue to be passed on and rolled even if there is someone in the middle attempting to eat the chocolate, if another 6 is rolled they take over from the person in the middle even if they have not had the chance to eat any chocolate If there are not many 6's being rolled you can add in a second number for students to look for to speed the game up e.g. both a 1 and a 6 will let them in the middle to attempt to eat the chocolate The game continues until there is no chocolate left to be eaten	DSH-teaching-presentation-starlink-unplugged.ppt – slides 27 - 29 or Starlink HTML5 - slides 27 - 29 DSH-party-game-flowcharts-starlink.pdf Dice Hat Gloves Scarf Wrapped chocolate bar Plate Knife Fork
	Explain that the Chocolate Game is an example of a game that uses nesting as there is an outer loop that continues all the time there is still chocolate to eat and an inner loop that starts once a 6 has been rolled: to put on a hat, gloves and scarf and then attempt to eat the chocolate until someone else gets a 6	DSH-teaching-presentation-starlink-unplugged.ppt – slides 27 - 29 or Starlink HTML5 - slides 27 - 29

For ADVANCED you could also discuss that this is an example of multi-threading (or parallel processing) as the loop that is looking for a 6 continues at the same as the loop for eating the chocolate	DSH-party-game-flowcharts-starlink.pdf
Read the slide that explains to students that the next task will be using the constructs to program a star ship!	DSH-teaching-presentation-starlink-unplugged.ppt – slide 30 or Starlink HTML5 – slide 30

The SCORM and HTML5 versions of the following activities use the PRIMM approach to teaching programming and include the predict stage as a self-marking multiple-choice question

SESSION 2

Activity No	Session Content / Activity	Resources Used
B Intro	In this section students will follow the step by step instructions to learn how to access Scratch 3 and save and load files	DSH-teaching-presentation-starlink-scratch.ppt – slides 1 – 10 or Starlink HTML5 – slides 38 - 43 or Starlink SCORM – slides 1 - 15
B1	In this section students will follow the step by step instructions to learn how to imports sprites into Scratch and program their ship to move.	DSH-teaching-presentation-starlink-scratch.ppt – slides 11 – 38 or Starlink HTML5 – slides 44 - 71 or Starlink SCORM – slides 16 - 43
I Intro	In this section students will follow the step by step instructions to learn how to access Construct 3 and save and load files	DSH-teaching-presentation-starlink-construct 3.ppt – slides 1 – 11

		or Starlink HTML5 – slides 105 - 111 or Starlink SCORM – slides 77 - 83
I1	In this section students will follow the step by step instructions to learn how to import sprites into Construct 3 and program their ship to move	DSH-teaching-presentation-starlink-construct 3.ppt – slides 12 – 38 or Starlink HTML5 – slides 112 - 138 or Starlink SCORM – slides 84 - 110
A Intro	In this section students will follow the step by step instructions to learn how to access Unity and save and load files	DSH-teaching-presentation-starlink-unity.ppt – slides 1 – 4 or Starlink HTML5 – slide 190 or Starlink SCORM – 162 or Unity_001_Introduction.mp4
A1	In this section students will follow the step by step instructions to learn how to import sprites into Unity	DSH-teaching-presentation-starlink-unity.ppt – slide 5 or Starlink HTML5 – slides 191 - 194 or Starlink SCORM – slides 163 - 166 or Unity_002_ResourcesAndBackground.mp4

SESSION 3

Activity No	Session Content / Activity	Resources Used
B2	In this section students will follow the step by step instructions to learn how to program the meteor to move in Scratch	DSH-teaching-presentation-starlink-scratch.ppt – slides 39 - 49 or Starlink HTML5 – slides 72 - 84 or Starlink SCORM – slides 44 - 56
I2	In this section students will follow the step by step instructions to learn how to program the meteor to move in Construct 3	DSH-teaching-presentation-starlink-construct 3.ppt – slides 39 - 54 or Starlink HTML5 – slides 139 - 155 or Starlink SCORM – slides 111 - 127
A2	In this section students will follow the step by step instructions to learn how to program the ship to move in Unity	DSH-teaching-presentation-starlink-unity.ppt – slide 6 or Starlink HTML5 – slides 195 - 197 or Starlink SCORM – slides 167 - 169 or

Unity_003_PlayerMovement.mp4

SESSION 4

Activity No	Session Content / Activity	Resources Used
B3	In this section students will follow the step by step instructions to learn how to add lasers that fire on the press of a key in Scratch	DSH-teaching-presentation-starlink-scratch.ppt – slides 50 - 58 or Starlink HTML5 – slides 85 - 93 or Starlink SCORM – slides 57 - 65
I3	In this section students will follow the step by step instructions to learn how to add lasers that fire on the press of a key in Construct 3	DSH-teaching-presentation-starlink-construct 3.ppt – slides 55 - 75 or Starlink HTML5 – slides 156 - 177 or Starlink SCORM – slides 128 - 149
A3	In this section students will follow the step by step instructions to learn how to program the meteor to move in Unity	DSH-teaching-presentation-starlink-unity.ppt – slide 7 or Starlink HTML5 – slides 198 - 200 or Starlink SCORM – slides 170 - 172

or
 Unity_004_Meteors.mp4

SESSION 5

Activity No	Session Content / Activity	Resources Used
B4	In this section students will follow the step by step instructions to learn how make the meteors explode on collision with the ships lasers in Scratch	DSH-teaching-presentation-starlink-scratch.ppt – slides 59 - 61 or Starlink HTML5 – slides 94 - 97 or Starlink SCORM – slides 66 - 69
I4	In this section students will follow the step by step instructions to learn how make the meteors explode on collision with the ships lasers in Construct 3	DSH-teaching-presentation-starlink-construct 3.ppt – slides 76 - 79 or Starlink HTML5 – slides 178 - 182 or Starlink SCORM – slides 150 - 154
A4	In this section students will follow the step by step instructions to learn how make the meteors explode on collision with the ships lasers in Unity	DSH-teaching-presentation-starlink-unity.ppt – slides 8 - 9 or Starlink HTML5 – slides 201 - 203 or

Starlink SCORM – slides 173 - 175

or

Unity_005_ExplosionEffect.mp3 & Unity_006_Lasers.mp4

SESSION 6

Activity No	Session Content / Activity	Resources Used
B5	In this next section students can demonstrate their understanding of what they have learnt by attempting 5 challenges in Scratch: Challenge 1 – Add sound Challenge 2 – Program the ship to move in all directions Challenge 3 – Add instructions to the start of the game Challenge 4 – Add a scoring system Challenge 5 – Add randomised meteor movement	DSH-teaching-presentation-starlink-scratch.ppt – slides 62 - 69 or Starlink HTML5 – slides 98 - 105 or Starlink SCORM – slides 70 - 76
I5	In this next section students can demonstrate their understanding of what they have learnt by attempting 5 challenges in Construct 3: Challenge 1 – Add sound Challenge 2 – Program the ship to move in all directions Challenge 3 – Add instructions to the start of the game	DSH-teaching-presentation-starlink-construct 3.ppt – slides 80 - 87 or Starlink HTML5 – slides 183 - 189 or Starlink SCORM – slides 155 - 161

	Challenge 4 – Add a scoring system	
	Challenge 5 – Add randomised meteor movement	
A5	In this next section students can demonstrate their understanding of what they have learnt by attempting 5 challenges in Unity: Challenge 1 – Add sound Challenge 2 – Program the ship to move in all directions Challenge 3 – Add instructions to the start of the game Challenge 4 – Add a scoring system Challenge 5 – Add a second level They will then follow the instructions to export their game	DSH-teaching-presentation-starlink-unity.ppt – slides 10 - 19 or Starlink HTML5 – slides 204 - 212 or Starlink SCORM – slides 176 - 184

The self-marking assessment for this workshop is only available in the SCORM and HTML5 versions of the tutorials

SESSION 7

Activity No	Session Content / Activity	Resources Used
Assessment	Students should complete the assessment (if completing the HTML5 version, students will be prompted to complete a teacher email and other details before starting the assessment – results will be sent to this email).	Starlink HTML5 – slides 220 - 222 or Starlink SCORM – slides 192 - 194

Students will be asked to select the assessment level they want to work at (Beginner, intermediate or advanced).

There are 10 self-assessment questions.

Digital Schoolhouse Progression Matrix

The Digital Schoolhouse progression matrix is a simplified mechanism for measuring pupil progress. It stretches from base level understanding at the beginner level to introducing GCSE content at the advanced level.

The shaded statements reflect skills and concepts covered in the workshop, these have been cross referenced to specific activities in order to reflect both their level of complexity and provide a guideline on which to measure progress.

For more details about this framework see 'Enter the Matrix' included in this workshop pack.

U = Unplugged activity, B = Beginner activity, I = Intermediate activity, A = Advanced activity

Algorithms

	Beginner	Activity No	Intermediate	Activity No	Advanced	Activity No
Understanding	Understands what an algorithm is	U1 – U5	Understands that algorithms are not the same as programming		Understands that different algorithms exist for the same problem	
Writing	Represents algorithms using graphical notation such as pictures		Represents algorithms using structured notation such as flowcharts		Represents algorithms using pseudocode	
Limitations	Understands that computers need precise instructions		Can identify tasks best completed by humans or computers		Understands that some problems cannot be solved computationally	
Planning	Can identify the steps needed to solve a problem		Can identify the programming constructs needed to solve a problem (pattern recognition)	U4, U5	Can identify the modules needed to solve a problem e.g. top down design	
Tracing	Uses logical reasoning to predict outputs and show an awareness of inputs		Uses logical reasoning to explain how an algorithm works		Evaluates the effectiveness of algorithms and models for similar problems	
Designing	Designs solutions (algorithms) that use sequence, selection i.e. if, then and else and iteration	U1 – U5	Designs solutions by decomposing a problem and creating a sub-solution for each of these parts		Designs a solution to a problem that uses generalization to create objects and classes (OOP)	

Programming

	Beginner	Activity No	Intermediate	Activity No	Advanced	Activity No
Writing	Can create a simple program in an environment that does not rely on text e.g. programmable robots etc	B1, B2, B3, B4, B5	Has practical experience of a high-level textual language, including use of standard libraries		Has experience of designing programs that include a graphical user interface	A2, A3, A4, A5
Program flow	Understands that programs execute by following precise instructions	U1 B2, B3, B4, B5	Understands how modular programs work using sub-routines		Appreciates the effect of the scope of a variable e.g. a local variable can't be accessed from outside its function unless passed as a parameter	A2, A3, A4, A5
Selection	Uses selection statements in programs, including an if, then and else statement	U2, U4, U5 B2, B3, B4, B5	Understands the difference between, and appropriately uses if and if, then and else Statements	U2 B2, B3, B4, B5	Uses nested selection statements	
Iteration	Uses loops, within programs	U3, U4, U5 B2, B3, B4, B5	Understands the difference between, and uses 'while', 'until' and 'for' loops	U3, U4, U5 B2, B3, B4, B5	Uses nested iteration and recursion	U5
Debugging	Detects and corrects simple semantic errors i.e. debugging, in programs	B2, B3, B4, B5	Detects and corrects syntactical errors	I2, I3, I4, I5	Applies a modular approach to error detection and correction	A2, A3, A4, A5
Program design	Creates programs that implement algorithms to achieve given goals	B2, B3, B4, B5	Can design a program based on an algorithm	I2, I3, I4, I5	Designs modular programs using a range of methodologies e.g. RAD, waterfall	A2, A3, A4, A5
Data types and structures	Declares and assigns variables	B5	Selects appropriate data types	I2, I3, I4, I5	Understands and uses one- and two-dimensional data structures	A5
Operators	Uses arithmetic operators	B2, B3, B4, B5	Uses a range of operators and expressions e.g. Boolean	I2, I3, I4, I5	Understands and uses negation with operators e.g. not equal to	A5

Data

	Beginner	Activity No	Intermediate	Activity No	Advanced	Activity No
Representation	Recognises that digital content can be represented in many forms	B2, B3, B4, B5	Understands how bit patterns represent different forms of data e.g. character sets, sound, numbers and images		Understands how the same bit patterns can be used for different forms of data e.g. metadata	
Transfer	Knows that data can be transferred from one computer to another		Knows that computers transfer data in binary		Understands and can explain the need for data compression, and performs simple compression methods	
Types	Recognises different types of data: text, number	B2, B3, B4, B5	Defines data types: string, integer, real and Boolean		Understands how different data types can be used within data structures e.g. arrays must be made up of the same data type whereas lists can use several	A2, A3, A4, A5
Binary	Can carry out simple binary to decimal conversions	B2, B3, B4, B5	Performs operations using bit patterns e.g. binary addition, conversion between binary and hexadecimal, binary subtraction etc		Understands the relationship between binary and electrical circuits, including Boolean logic	
File Size	Understands that data takes up space on a computer		Understands the relationship between binary and file size (uncompressed)		Knows the relationship between data representation and data quality e.g. resolution and colour depth etc, including the effect on file size	
Data and Information	Understands the difference between data and information		Recognises that poor-quality data leads to unreliable results, and inaccurate conclusions	I2, I3, I4, I5	Understand the mechanisms used to cleanse data e.g. validation, range checks etc	
Searching	Can sort data, use filters and perform single criteria searches for information		Queries data on one table using a typical query language, including more complex searches for information e.g. using Boolean and relational operators		Queries data on multiple tables using a typical query language	
Structure	Recognises that data can be structured in tables to make it useful		Understands that all the data about a person or thing can be stored as a record		Knows what a relational database is, and understands the benefits of storing data in multiple tables	

Hardware and Software

	Beginner	Activity No	Intermediate	Activity No	Advanced	Activity No
Processing	Understands that computers have no intelligence and that computers can do nothing unless a program is executed	B2, B3, B4, B5	Knows that programs are executed by the processor i.e. the CPU		Understand that processors can work in different ways e.g. multitasking, scheduling	
Software	Recognises that all software executed on digital devices is programmed	B2, B3, B4, B5	Knows that there is a range of operating systems and application software for the same hardware		Understands the concept of proprietary and open-source software including how this relates to licencing	
Devices	Recognises that a range of digital devices can be considered a computer		Understands why and when computers are used	I1	Understands how technology has developed e.g. Moore's Law	
Components	Recognises and can use a range of input and output devices		Recognises and understands the function of the main internal parts of basic computer architecture		Knows that processors have instruction sets and that these relate to low-level instructions carried out in the main internal parts of a computer	
Operating systems	Understands that the operating system is software that specifies the function of a computing device		Understands the main functions of the operating system		Understands that there are different types of operating system and some of their common uses e.g. real time on auto pilot systems on a plane	
Data transfer	Knows that data is transferred around a computer system using input devices, sensors and application software	B2, B3, B4, B5	Knows that data can be transferred between computer systems using physical, wireless and mobile networks		Knows how data can be transferred between computer systems e.g. packet and circuit switching	
Architecture	Understands the difference between hardware and software		Understands how hardware uses software to execute instructions e.g. the fetch-execute cycle		Understands computer architecture in relation to the fetch execute cycle, including how data is stored in memory	

Communication

	Beginner	Activity No	Intermediate	Activity No	Advanced	Activity No
WWW	Accesses content using a web browser		Understands that web pages are created using HTML and CSS		Understands how dynamic web pages use the client-server model and that web servers process and store data entered by users	
Online safety	Understands why and how to keep personal information private and knows what to do when concerned about something online		Has an awareness of a range of online harms and demonstrates responsible use of technologies and online services in order to protect themselves		Understands how and why online threats are carried out and how to protect against them	
Search engines	Navigates the web and can carry out simple web searches to collect digital content		Understands how to effectively use search engines e.g. Boolean, advanced search functions etc		Knows how search results are selected and ranked, including that search engines use 'web crawler programs'	
Networks	Understands the difference between the internet and internet service e.g. world wide web		Understands data is transmitted between digital computers over networks, including different topologies e.g. ring, star, mesh		Knows the names and purposes of network components and protocols	
Internet services	Shows an awareness of, and can use a range of internet services e.g. email		Selects, combines and uses internet services		Uses internet services to work collaboratively	

IT

	Beginner	Activity No	Intermediate	Activity No	Advanced	Activity No
Invention	Uses software under the control of the teacher to create, store and edit digital content	B1, B2, B3, B4, B5	Uses and selects internet services, digital devices and application software to create, store and edit digital content	I1, I2, I3, I4, I5	Evaluates the appropriateness of digital devices, internet services and application software to achieve given goals	
Audience	Understands what an audience is	B1, B2, B3, B4, B5	Recognises the audience when designing and creating digital content		Undertakes creative projects that are tailored to meet the needs of an audience	A1, A2, A3, A4, A5
Purpose	Can talk about how they use computers		Can talk about how other people use computers		Can discuss the issues around how other people might use computers e.g. Data Protection Act, Computer Misuse Act, Copyright etc	
Evaluation	Can comment on the success of their solution		Designs and uses criteria to critically evaluate the quality of solutions		Documents user feedback, the improvements identified, and the refinements made to the solution	
Content	Can gather content	B2	Makes judgements about content when evaluating and repurposing it for a given audience	I2	Evaluates the trustworthiness of content, considers the usability of visual design features and properties of media when designing and creating digital artefacts	A2

Computing Programmes of Study Links

- 1.1 understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- 1.2 create and debug simple programs
- 1.3 use logical reasoning to predict the behaviour of simple programs
- 1.4 use technology purposefully to create, organise, store, manipulate and retrieve digital content
- 1.5 recognise common uses of information technology beyond school
- 1.6 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

- 2.1 design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- 2.2 use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- 2.3 use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- 2.6 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
- 2.7 use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact

- 3.1 design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
- 3.3 use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions
- 3.7 undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users
- 3.8 create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability
- 3.9 understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns

- 4.1 develop their capability, creativity and knowledge in computer science, digital media and information technology
- 4.2 develop and apply their analytic, problem-solving, design, and computational thinking skills

Computational Thinking Strands

AL – Algorithmic Thinking

Ref. Activity

A1	Formulating instructions to achieve a desired effect
A2	Formulating instructions to be followed in a given order (sequence)
A3	Formulating instructions that use arithmetic and logical operations
A4	Writing sequences of instructions that store, move and manipulate data (variables and assignment)
A5	Writing instructions that choose between different constituent instructions (selection)
A6	Writing instructions that repeat groups of constituent instructions (loops/iteration)
A7	Grouping and naming a collection of instructions that do a well-defined task to make a new instruction (subroutines, procedures, functions, methods)
A11	Using an appropriate notation to write code to represent any of the above
A12	Creating algorithms to test a hypothesis

DE – Decomposition

Ref. Activity

D1	Breaking down artefacts into constituent parts to make them easier to work with
D2	Breaking down a problem into simpler versions of the same problem that can be solved in the same way (recursive and divide and conquer strategies)

GE – Generalisation

Ref. Activity

- | | |
|----|---|
| G1 | Identifying patterns and commonalities in artefacts |
| G2 | Adapting solutions, or parts of solutions, so they apply to a whole class of similar problems |
| G3 | Transferring ideas and solutions from one problem area to another |

AB – Abstraction

Ref. Activity

- | | |
|-----|---|
| Ab1 | Reducing complexity by removing unnecessary detail |
| Ab2 | Choosing a way to represent an artefact, to allow it to be manipulated in useful ways |
| Ab3 | Hiding the full complexity of an artefact (hiding functional complexity) |

EV – Evaluation

Ref. Activity

- | | |
|-----|---|
| E1 | Assessing that an artefact is fit for purpose |
| E2 | Assessing whether an artefact does the right thing (functional correctness) |
| E4 | Assessing whether the performance of an artefact is good enough (utility: effectiveness and efficiency) |
| E10 | Stepping through processes or algorithms/code step-by-step to work out what they do (dry run/tracing). |