



**DIGITAL
SCHOOLHOUSE**
together with



+44 (0) 207 534 0586
dsh@ukie.org.uk

Teaching Guide.

Part-Baked Games

Introduction

Often, learning how to create a game can be quite daunting, especially when you consider all the elements that are needed to build it. Part-Baked Games provides the learner with the opportunity to create five different games that are inspired by the BAFTA YGD rule cards in order to teach learners how to build common game mechanics and therefore provide them with the tools needed to create their own game at a later date. Each of the games are provided in a 'Part-Baked' form which means that all assets required for the game have been added and laid out as though the game were complete. Learners simply follow the instructions to add the game mechanics needed to complete the game.

Part-Baked Games makes an excellent introductory workshop to deliver before Loopy Games or as an introduction to the skills needed to enter the BAFTA YGD competition. As with all Digital Schoolhouse workshops, Part-Baked Games includes an unplugged section where learners can discover the different game mechanics they will be learning about through the power of play.

ukie



PlayStation.

SEGA®



Warwickshire
County Council



UBISOFT

Contents

Learning Outcomes	3
Files/Resources.....	4
Session Overview.....	6
Digital Schoolhouse Progression Matrix.....	18
Algorithms	18
Programming	19
Data	20
Hardware and Software.....	21
Communication	22
IT.....	23
Computing Programmes of Study Links.....	24
Computational Thinking Strands	26

Learning Outcomes

1. To be able to identify different game mechanics
2. To be able to explain the importance of correct instructions
3. To be able to explain what an algorithm is
4. To be able to implement a pre-written algorithm within a game engine such as Construct 3
5. To be able to explain how a variable can be used
6. To be able to create and use variables within a game
7. To be able to create and use behaviours within a game
8. To be able to create and use events within a game
9. To be able to recognise what triggers an event and the action that will be taken as a result of it
10. To be able to dry run/test their game to see if it works and achieves desired results
11. 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-Teaching-Presentation-Part-Baked-Games.ppt	Teacher presentation	Teaching presentation for Part-Baked Games workshop Detailed step-by-step instructions are included in this presentation	All activities
DSH-Worksheets-Part-Baked-Games.ppt	Worksheets	Worksheets for Part-Baked Games workshop Included in this pack are Part-Baked Game cards which provide simple instructions for creating each game mechanic	U1, U12, 13, 14, 15, 16, 17
Newspaper (1 sheet per learner)	Activity resource	Needed for The Newspaper game	U3
Music	Activity resource	Needed for Musical Statues, The Newspaper Game and Musical Chairs	U3, U4, U8
Large bunch of keys	Activity resource	Needed for keeper of the Keys	U5
Access to a large empty hall	Activity resource	Needed for King Ball	U6
Bouncy ball	Activity resource	Needed for King Ball	U6
8 random everyday items (e.g. glasses, pen, phone etc.)	Activity resource	Needed for Memory Game	U8
Scrap paper	Activity resource	Needed for Memory Game	U8
Chairs (1 per person)	Activity resource	Needed for Keeper or the Keys and Musical Chairs	U5, U9
Timer	Activity resource	Needed for Memory Game and Pictionary	U8, U11
Large piece of paper			U11

	Activity resource	Needed for Pictionary	
Pictionary cards or a list of simple objects to draw	Activity resource	Needed for Pictionary	U11
Pencil or pens (one per learner)	Activity resource	Needed for identifying the game mechanics and completing worksheets	U1, U8, U11, U12
PC or laptop	Activity resource	Needed for creating all Part-Baked games	All Part-Baked Games
Google Chrome	Software	Needed to access Construct 3 for creating games	All Part-Baked Games
Construct 3 (downloaded as an app or run in browser) https://editor.construct.net/	Software	Needed to complete Part-Baked Games	All Part-Baked Games
RunTheGauntletPartBaked.c3p	Part-Baked Game	Needed for creating Run the Gauntlet	13
TheFloorIsLavaPartBaked.c3p	Part-Baked Game	Needed for creating The Floor is Lava	14
PipesPartBaked.c3p	Part-Baked Game	Needed for creating Pipes	15
ReleaseTheMonstersPartBaked.c3p	Part-Baked Game	Needed for creating Release the Monsters	16
CaptureThePigsPartBaked.c3p	Part-Baked Game	Needed for creating Capture the Pigs	17

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

SESSION 1 - UNPLUGGED

Activity No	Session Content / Activity	Resources Used
	Welcome and introduction	DSH-Teaching-Presentation-Part-Baked-Games – Slides 1 and 2
U1	<p>Read the slide giving the definition of Game Mechanics</p> <p>Move onto the next slide and explain that there are lots of different game mechanics but we will be focusing on 10 – read the slide to your learners</p> <p>Move to the next slide and then explain that all computer games use game mechanics, hand out mind-map worksheets and encourage students to identify game mechanics that are used in their favourite game – you could put slide 4 back up to help give them ideas (they do not need to be limited to these)</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games – Slides 3 – 5</p> <p>DSH-Worksheets-Part-Baked-Games – Slide 2 – printed, one copy per learner Pens or pencils (1 per person)</p>
U2	<p>Explain that all games have game mechanics and we can identify some of them in playground games</p> <p>Move to the next slide and explain that they are going to play a game of IT</p> <p>Play a game of IT: one learner is IT and the rest of the learners must avoid being touched by this person. If someone is touched by IT they are immediately out and have to move to the side of the room. The game continues until there are no people left to catch.</p>	DSH-Teaching-Presentation-Part-Baked-Games – Slides 6 - 7

	Once the game has been completed ask students what game mechanic they think it uses – they should identify that it uses the Avoid Enemies game mechanic, encourage them to come to the conclusion that the people were avoiding their enemy – the IT person	
	Show the next slide and explain that we are going to be playing some games and that while the learners are playing, they should be thinking about the game mechanics they are using	DSH-Teaching-Presentation-Part-Baked-Games – Slide 8
U3	Play a game of The Newspaper Game: Each learner should place their sheet of newspaper on the floor and stand on it. The newspaper sheets should be spread around the room and none should overlap. Remove one sheet of paper so there is 1 less sheet than the number of players. When the music plays all learners should move around the room and when it stops, they must move quickly to stand on a sheet of newspaper. The player who is not on a sheet is out. The game continues until only one player remains and they are announced the winner!	DSH-Teaching-Presentation-Part-Baked-Games – Slide 9 Music Sheets of newspaper (1 per player)
U4	Play a game of Musical Statues: Each learner should dance while the music plays. If the music stops, they must strike a pose. They must not do the same pose as they did in the previous round. The person who is last to pose or who repeats a pose is out. Repeat until there is one player left – they are the winner!	DSH-Teaching-Presentation-Part-Baked-Games – Slide 10 Music
U5	Play a game of Keeper of the Keys: Choose one player, they should sit on a chair with the keys placed below the chair. The rest of the players should sit on the floor at the other end of the room. The player on the chair should have their eyes closed (they can be blindfolded). One at a time give the rest of the players the chance to creep up on the player on the chair and take the keys without being caught. The player on the chair should listen carefully and point if they think they can hear a player coming. If the creeping player is correctly pointed at then they are out and return to their place. The game repeats until a player creeps up and gets the keys without being 'seen'	DSH-Teaching-Presentation-Part-Baked-Games – Slide 11 Large set of keys Chair
U6	This game is best played in a hall	DSH-Teaching-Presentation-Part-Baked-Games – Slide 12

	<p>Play a game of King Ball: Choose one player and give them the ball. The rest of the players should move around in the space in the middle of the room. The player with the ball should attempt to bounce the ball off the wall and hit one of the players in the middle (the middle players should try to dodge the ball). If a player is hit they are out and should leave the play area. The game ends when there are no people left in the middle.</p>	<p>Large empty hall Bouncy ball</p>
U7	<p>Play a game of Doctor Doctor: To make this game easier, it can be played in two smaller groups. Nominate a 'Doctor' they should stand to one side. The rest of the players stand in a circle and put both hands into the middle. Each person should take hold of two different people's hands. They can then call the 'Doctor' who should attempt to untangle them without breaking the circle. The players should do as the 'Doctor' asks but should not let go of the hands they are holding. The game ends when the circle is untangled.</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games – Slide 13</p>
U8	<p>Play a game of the Memory Game: Place the 8 items on a table, all players have a minute to memorise the items on the table – the items are then covered up and the players have to list as many items as they can remember. The winner is the player who can remember the most.</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games – Slide 14</p> <p>8 random everyday items (e.g. glasses, pen, phone etc.) Scrap paper Pens or pencils (1 per person)</p>
U9	<p>Play a game of Musical Chairs: Place all the chairs in a row with each alternate chair facing the opposite direction. Remove one chair so there is one less chair than the number of players. When the music plays, the players should move around the chairs and when the music stops, they must proceed to the nearest chair and sit down, but they must only move forward. The player unable to find a chair or any players that go backwards are out. Remove another chair each round until only one player remains and the last player standing is the winner!</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games – Slide 15</p> <p>Music Chairs (1 per player)</p>
U10	<p>Play a game of Wink Murder: Have the learners sit in a circle, nominate one learner as the detective and send them into the corridor. Secretly choose a learner left in the circle as the murderer – make sure that</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games – Slide 16</p>

	<p>everyone (but not the detective) knows who the murderer is. Invite the murderer back, they should carefully watch as the game continues and try to guess who the murderer is – they have 3 guesses.</p> <p>The murderer should murder people by winking at them, if a player is winked at, they should die an over dramatic and loud death. If the murderer murders all players before being caught or if all 3 guesses have been used up the murderer wins. If the detective guesses correctly then they win. The game ends when a winner is identified.</p>	
U11	<p>Play a game of Pictionary: Nominate one person to be the player, they should look at a Pictionary card (or one of the items from the list) and draw it for the class. They will have a time limit of 1 minute. If the picture is guessed, then they can move onto a new card. When the time runs out the player should be swapped and another player chosen. The winner is the person with the most correctly identified drawings in a minute.</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games – Slide 17</p> <p>Pictionary cards or a list of things to draw Pen or pencil Large piece of paper Timer</p>
U12	<p>Read students the next slide and explain that they are going to identify the game mechanics used for each playground game</p> <p>Hand out the worksheet and a pen or pencil each</p> <p>Move onto the next slide so that learners can be reminded about the names of the mechanics that they have used</p> <p>Give students time to fill in the grid on their worksheet</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games – Slides 18 – 19</p> <p>DSH-Worksheets-Part-Baked-Games.ppt – slide 3 – printed 1 per person Pens or pencils (1 per person)</p>
U12	<p>Spend some time going over the answers – each slide explains one game mechanic that is used for the game</p> <p>Students may have identified different game mechanics – this is fine – encourage students to explain their choices</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games – Slides 20 - 28</p>

SESSION 2

Activity No	Session Content / Activity	Resources Used
	Move onto the Part-Baked Games section and read the slides that explain what Part-Baked games are and how they will be used.	DSH-Teaching-Presentation-Part-Baked-Games – Slides 29 - 31
13	<p>Show slide 32 - Run the Gauntlet</p> <p>Show slides 33 - 34 and make sure that all students have RunTheGauntletPartBaked opened in Construct 3</p> <ol style="list-style-type: none"> 1. Access https://editor.construct.net/ 2. Click on File and open the RunTheGauntletPartBaked.c3p file <p>Show slide 35 and make sure students know how to save their work – you may need to explain how to move work from the downloads folder dependent on computer setup</p> <p>Show slide 36 – 37 and give students some time to explore the games assets.</p> <p>Show slides 38 – 39 and give students some time to test the Part-Baked version of the game. Ask students what game mechanics they think are missing from the game. Encourage some discussion here.</p> <p>Show slide 40 and explain that learners will be learning how to implement the following mechanics: can only move forwards, bounce off walls and avoid enemies.</p> <p>Work through slides 41 - 86, make sure you highlight each of the mechanics (there are slides to point out where these instructions start) and give students some time to tinker with adding the mechanics themselves before moving onto demonstrating one way of adding them. If students are successful in adding mechanics we would recommend encouraging students to demonstrate this for their peers.</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games – Slides 32 - 86</p> <p>PC or laptop</p> <p>Google Chrome</p> <p>Construct 3 (downloaded as an app or run in browser): https://editor.construct.net/</p> <p>RunTheGauntletPartBaked.c3p</p> <p>DSH-Worksheets-Part-Baked-Games – slide 4 – printed 1 copy per person</p>

There are several points throughout the presentation where learners are encouraged to run their game and test it – there are notes included on these slides to indicate how to lead discussion

If you have students that are able to work more independently, have some copies of the Part-Baked game card for Run the Gauntlet ready to hand out to students. This will allow you to focus the step-by-step instructions on learners that need more support.

SESSION 3

Activity No	Session Content / Activity	Resources Used
14	<p>Show slide 87 and explain how learners should used the Part-Baked game cards</p> <p>Show slide 88 – The Floor is Lava</p> <p>Show slide 89 – Give some time for students to play test the Part-Baked version of the game and identify what mechanics are missing.</p> <p>Show slide 90 - explain that learners will be learning how to implement the following mechanics: can't touch the floor and one life only.</p> <p>Show slide 91 – explain how X and Y coordinates work in the events</p> <p>Show slide 92 – explain how mirroring works</p> <p>Show slide 93 – explain how named animations work</p> <p>Show slide 94 - Give some time for students to tinker with adding the game mechanics themselves – any that are successful can demonstrate for the class</p> <p>Show slide 95 - hand out the Part-Baked game cards for The Floor is Lava and give learners time to complete the game – make sure that all students have opened TheFloorIsLavaPartBaked.c3p in Construct 3 (students should be able to work independently using the game card but a teacher should be available to support where needed)</p> <p>Show slide 96 - 97 – remind learners that they have completed a game in which the player can't touch the floor and they have one life only, ask students to think about how these have been added to the game. Encourage feedback and discussion from the class about how they think they have done this and explain how the mechanics were added</p> <p>Show slide 98 and celebrate the completed game!</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games – Slide 87</p> <p>DSH-Teaching-Presentation-Part-Baked-Games – Slides 88 - 98</p> <p>PC or laptop</p> <p>Google Chrome</p> <p>Construct 3 (downloaded as an app or run in browser): https://editor.construct.net/</p> <p>TheFloorIsLavaPartBaked.c3p</p> <p>DSH-Worksheets-Part-Baked-Games – slide 5 – printed 1 copy per person</p>

15

Show slide 99 – Pipes

Show slide 100 – Give some time for students to play test the Part-Baked version of the game and identify what mechanics are missing. Make sure that all students have opened PipesPartBaked.c3p in Construct 3, notes are included with the slide for discussion starters

Show slide 101 - explain that learners will be learning how to implement the following mechanics: limited time

Show slides 102 - 105 – explain how global variables differ from instance variables and how to add them in Construct 3

Show slide 106 – explain how system conditions differ from object conditions

Show slide 107– explain how to add multiple conditions

Show slides 108 – 109 – explain how to display the value of a variable in game

Show slide 110 – give your students some time to tinker with adding the mechanics themselves and encourage any that are successful to demonstrate for the class.

Show slide 111 - hand out the Part-Baked game cards for Pipes and give learners time to complete the game – make sure that all students have opened PipesPartBaked.c3p in Construct 3 (students should be able to work independently using the game card but a teacher should be available to support where needed)

Show slide 112 – remind learners that they have completed a game in which the player has a limited time to complete a task, ask students to think about how this has been added to the game. Encourage feedback and discussion from the class about how they think they have done this and then explain how the mechanic was added

Show slide 113 and celebrate the completed game!

DSH-Teaching-Presentation-Part-Baked-Games – Slides 99 - 113

PC or laptop

Google Chrome

Construct 3 (downloaded as an app or run in browser):
<https://editor.construct.net/>

PipesPartBaked.c3p

DSH-Worksheets-Part-Baked-Games – slide 6 – printed 1 copy per person

SESSION 4

Activity No	Session Content / Activity	Resources Used
16	<p>Show slide 114 – Release the Monsters</p> <p>Show slide 115 – Give some time for students to play test the Part-Baked version of the game and identify what mechanics are missing. Make sure that all students have opened ReleaseTheMonstersPartBaked.c3p in Construct 3, notes are included with the slide for discussion starters</p> <p>Show slide 116 - explain that learners will be learning how to implement the following mechanics: limited inventory, must not be seen and avoid enemies</p> <p>Show slides 117 – 119 – explain how negation can be used in Construct 3</p> <p>Show slide 120 – explain the concept of ‘every tick’</p> <p>Show slide 121– explain the advantages of using families, notes are included on the slide to aid with discussion</p> <p>Show slide 122 – give students some time to tinker with adding the mechanics themselves</p> <p>Show slide 123 - hand out the Part-Baked game cards for Release the Monsters and give learners time to complete the game – make sure that all students have opened ReleaseTheMonstersPartBaked.c3p in Construct 3 (students should be able to work independently using the game card but a teacher should be available to support where needed)</p> <p>Show slide 124- 126 – remind learners that they have completed a game in which the player has a limited inventory, must not be seen and must avoid enemies, ask students to think about how these have been added to the game. Encourage feedback and discussion from the class about how they think they have done this and then explain how the mechanics were added.</p> <p>Show slide 127 and celebrate the completed game!</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games – Slides 114 – 127</p> <p>PC or laptop</p> <p>Google Chrome</p> <p>Construct 3 (downloaded as an app or run in browser): https://editor.construct.net/</p> <p>ReleaseTheMonstersPartBaked.c3p</p> <p>DSH-Worksheets-Part-Baked-Games – slides 7 & 8 – printed 1 copy per person</p>

17

Show slide 128 – Capture the Pigs

Show slide 129 – Give some time for students to play test the Part-Baked version of the game and identify what mechanics are missing. Make sure that all students have opened CaptureThePigsPartBaked.c3p in Construct 3, notes are included with the slide for discussion starters

Show slide 130 - explain that learners will be learning how to implement the following mechanics: can't do the same thing twice in a row and can only select connecting items

Show slide 131 – explain how toggle works with Boolean values

Show slide 132 – explain how behaviours can be enabled and disabled in events – make sure students know that to use the disable behaviour they need to look for set enabled and then change the setting to disabled

Show slide 133 – give your students some time to tinker with adding the mechanics to their games. If any are successful these students should be given the opportunity to demonstrate to the class.

Show slide 134 – hand out the Part-Baked game cards for Capture the Pigs and give learners time to complete the game – make sure that all students have opened CaptureThePigsPartBaked.c3p in Construct 3 (students should be able to work independently using the game card but a teacher should be available to support where needed)

Show slide 135 - 136 – remind learners that they have completed a game in which the player can't do the same thing twice in a row and can only select connecting items, ask students to think about how these have been added to the game. Encourage feedback and discussion from the class about how they think they have done this and then explain how the mechanics were added

Show slide 137 and celebrate the completed game!

DSH-Teaching-Presentation-Part-Baked-Games – Slides 128 – 137

PC or laptop

Google Chrome

Construct 3 (downloaded as an app or run in browser):
<https://editor.construct.net/>

CaptureThePigsPartBaked.c3p

DSH-Worksheets-Part-Baked-Games – slide 9 – printed 1 copy per person

Remind students of the challenge set out at the beginning of the workshop and ask whether they think they have completed the challenge

DSH-Teaching-Presentation-Part-Baked-Games – Slides 138 - 139

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

SESSION 5

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). There are 10 self-assessment questions.	DSH-SCORM-Part-Baked-Games Or DSH-HTML5-Part-Baked-Games

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	All activities	Understands that algorithms are not the same as programming	All activities	Understands that different algorithms exist for the same problem	13, 14, 15, 16, 17
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	13, 14, 15, 16, 17	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	13, 14, 15, 16, 17	Can identify the programming constructs needed to solve a problem (pattern recognition)	13, 14, 15, 16, 17	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	13, 14, 15, 16, 17	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		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	13, 14, 15, 16, 17	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	
Program flow	Understands that programs execute by following precise instructions	13, 14, 15, 16, 17	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	
Selection	Uses selection statements in programs, including an if, then and else statement	13, 14, 15, 16, 17	Understands the difference between, and appropriately uses if and if, then and else Statements	13, 14, 15, 16, 17	Uses nested selection statements	16
Iteration	Uses loops, within programs	17	Understands the difference between, and uses 'while', 'until' and 'for' loops	17	Uses nested iteration and recursion	
Debugging	Detects and corrects simple semantic errors i.e. debugging, in programs	13, 14, 15, 16, 17	Detects and corrects syntactical errors	13, 14, 15, 16, 17	Applies a modular approach to error detection and correction	
Program design	Creates programs that implement algorithms to achieve given goals	13, 14, 15, 16, 17	Can design a program based on an algorithm		Designs modular programs using a range of methodologies e.g. RAD, waterfall	
Data types and structures	Declares and assigns variables	13, 14, 15, 16, 17	Selects appropriate data types	13, 14, 15, 16, 17	Understands and uses one and two dimensional data structures	
Operators	Uses arithmetic operators	13, 14, 15, 16, 17	Uses a range of operators and expressions e.g. Boolean	13, 14, 15, 16, 17	Understands and uses negation with operators e.g. not equal to	16

Data

	Beginner	Activity No	Intermediate	Activity No	Advanced	Activity No
Representation	Recognises that digital content can be represented in many forms	13, 14, 15, 16, 17	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	13, 14, 15, 16, 17	Defines data types: string, integer, real and Boolean	13, 14, 15, 16, 17	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	
Binary	Can carry out simple binary to decimal conversions		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		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	13, 14, 15, 16, 17	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	13, 14, 15, 16, 17	Knows that there is a range of operating systems and application software for the same hardware	13, 14, 15, 16, 17	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	13, 14, 15, 16, 17	Understands how technology has developed e.g. Moore's Law	
Components	Recognises and can use a range of input and output devices	13, 14, 15, 16, 17	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 these 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		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	13, 14, 15, 16, 17	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	13, 14, 15, 16, 17	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	13, 14, 15, 16, 17	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	13, 14, 15, 16, 17	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	13, 14, 15, 16, 17	Uses and selects internet services, digital devices and application software to create, store and edit digital content	13, 14, 15, 16, 17	Evaluates the appropriateness of digital devices, internet services and application software to achieve given goals	
Audience	Understands what an audience is		Recognises the audience when designing and creating digital content		Undertakes creative projects that are tailored to meet the needs of an audience	
Purpose	Can talk about how they use computers	13, 14, 15, 16, 17	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	13, 14, 15, 16, 17	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		Makes judgements about content when evaluating and repurposing it for a given audience		Evaluates the trustworthiness of content, considers the usability of visual design features and properties of media when designing and creating digital artefacts	

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

- 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

- 3.1 design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
- 3.2 understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem
- 3.3 use two or more programming languages, at least one of which is textual, to solve a variety of computational problems
- 3.4 understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in programming
- 3.5 understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems
- 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

- 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) |

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 |

EV – Evaluation

Ref. **Activity**

- | | |
|----|---|
| E1 | Assessing that an artefact is fit for purpose |
| E2 | Assessing whether an artefact does the right thing (functional correctness) |
| E3 | Designing and running test plans and interpreting the results (testing) |
| E4 | Assessing whether the performance of an artefact is good enough (utility: effectiveness and efficiency) |
| E8 | Assessing whether an artefact gives an appropriately positive experience when used (user experience) |
| E9 | Assessment of any of the above against the specification and set criteria |