



Declarative and Procedural Knowledge

Year 6

Contents

Introduction	3
Introduction to Purple Mash.....	4
Networks.....	5
Graphing	6
Blogging	7
Data Detectives.....	8
Coding.....	9
Coding Crash Course	11
Introduction to Python	15
Spreadsheets (Microsoft, Google and Apple)	16
3D Modelling.....	18
Binary	20
micro:bits.....	21

Introduction

The Declarative and Procedural Knowledge documents are designed to support teachers in understanding the intended learning outcomes of each unit. They outline the specific knowledge and skills that children should acquire and demonstrate by the end of their learning.

- Declarative Knowledge sets out what children will **know**. This includes facts, concepts, definitions, and key ideas that form the foundation of the unit.
- Procedural Knowledge sets out what children will **be able to do**. This focuses on the skills and processes children should develop and apply when using technology.

These documents are used to:

- Provide teachers with a clear overview of learning expectations for each unit.
- Ensure consistency of teaching and progression of knowledge and skills across year groups.
- Support planning, teaching, and assessment by highlighting the essential outcomes to focus on.
- Reinforce the balance between understanding (knowing) and application (doing) in computing.

This document aims to help teachers see the bigger picture of what children will learn, how they will apply it, and how these elements connect across the computing curriculum.

Introduction to Purple Mash

National Curriculum Links	Dominant objectives for this unit: 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.
---	---

Declarative - By the end of the unit, the children will know that:	Procedural – By the end of the unit, the children will know how to:
It is important to log in to a site, the importance of keeping passwords safe and the need to log out at the end of a session.	- Access Purple Mash from home and school. - Log out of Purple Mash. - Give reasons why it is important to keep a password safe and not share it with other people.
An avatar is a virtual representation of a person suitable for use online.	Make and edit their own avatar.
The 2Do system is used to set work for children within Purple Mash.	- Open 2Dos. - Save 2Dos. - Hand in 2Dos and communicate with their teacher via the 2Do.
Online sites have a main page called the homepage.	Access the Purple Mash homepage when on the site.
Online sites often use an alert system to communicate with the user.	Access alerts within Purple Mash.
To move to a different activity in Purple Mash, you must close the current activity.	Close activities in Purple Mash.
Many online sites, including Purple Mash, have an area for an individual's work that is accessible only to the individual (and in Purple Mash to their teacher as well).	- Access their work area. - Save work in their work area. - Locate and open work they have done previously in their work folder.
To access Purple Mash programs, you use the Tools area.	Open a specified tool.
To access activities related to a specific topic, you can use the Topics area.	Find activities on a specified topic.
You can access non-visible parts of a screen using scrolling.	Scroll up and down and from side to side where applicable.
Purple Mash includes collaborative tools.	Recommend a tool to use for collaborative group or class work.

Need more support? Contact us:

Call: 020 8203 1781 | Email: support@2simple.com | Visit: help.2simple.com

Powered by 

Networks

National Curriculum Links	Dominant objectives for this unit: 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.
----------------------------------	--

Declarative - By the end of the unit, the children will know that:	Procedural – By the end of the unit, the children will know how to:
A network describes a group of connected computers that can share information and hardware resources.	- Identify types of computer networks locally and globally. - Explain the hardware resources that a network might share.
LAN and WAN are different kinds of networks.	Explain the difference between LAN and WAN.
Certain hardware is required to create a network.	Create a network diagram that includes hardware such as a router and different connected devices and peripherals.
Networks can be wired or wireless or a combination of both.	Identify the terms Wi-Fi, mobile data and 5G as pertaining to wireless network connections.
There is a difference between the World Wide Web and the Internet.	Describe the difference between the Internet and World Wide Web giving examples of the services that both provide.
Web browsers are used to access the World Wide Web.	Give examples of web browser tools.
The existence of networks has opened online communication.	- Give examples of online communication. - Give safety tips related to online communication
Internet filtering and censorship are both used to make parts of the internet less accessible for different reasons.	Explain the differences between internet filtering and censorship and why they are used.

Graphing

National Curriculum Links	Dominant objectives for this unit: 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.
----------------------------------	---

Declarative - By the end of the unit, the children will know that:	Procedural – By the end of the unit, the children will know how to:
Graphing helps to make sense of datasets and draw conclusions related to the collected data	Create a variety of graphs and interpret these to draw conclusions.
There are different types of graphs. The data and the question that needs answering will determine the best graph type to produce.	Create a variety of graph types and determine the best format to represent specified data.
Comparative bar charts can be used to visually compare several datasets.	- Create a comparative bar chart using the 2Graph tool. - Present the graph with a title, key and axis labels.
Graphs can be exported from a graphing tool such as 2Graph and imported into other documents.	Export graphs from 2Graph and import them into a 2Publish file.
Pie charts represent data as parts of a whole.	- Use 2Graph and 2Calculate to create pie charts and then interpret what they show. - Compare the use of each tool in relation to graph production.
Line graphs are used to represent the relationship between two variables as they change over time.	- Decide when a line graph would be the most appropriate graphing format. - Create line graphs and use a graphing tool to add titles, labels and select the best scale for display. - Create line graphs showing multiple datasets and use these to draw conclusions about the data.

Blogging

National Curriculum Links	Dominant objectives for this unit: • 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.
---	--

Declarative - By the end of the unit, the children will know that:	Procedural – By the end of the unit, the children will know how to:
• A blog is a regularly updated webpage, written about a particular topic.	• Give examples of topics for existing or prospective blogs.
• Blogs consist of several blog posts.	• Create a blog post.
• A well written blog post has certain features that make the blog clear and easy to understand and increase reader engagement.	• Plan the hook, look and feel, conclusion and reader engagement for a blog post.
• The ‘hook’ draws the reader into the blog.	• Use an appropriate hook for a blog post by including either a quote, a story, a question or an observation to grab the reader’s interest.
• The look and feel of a blog post makes it clear for the reader to access the information.	• Write a blog post that is easy to follow, uses lists or bullets, bolds key information and uses an appropriate conversational style.
• The conclusion of a blog post ties the information in the post together.	• Write a conclusion that summarises the main points of the post and might give the reader advice.
• The process of writing a blog post requires planning, drafting, revising and editing before publication.	• Follow the plan, draft, revise and edit process before publishing a blog post.
• Engaging with readers is crucial to the success of a blog.	• Read and respond to comments on their blog post. • Use commenting to increase engagement and guide future blog posts.
• Moderation exists to make the blogging environment a safe place for its readership and authors.	• Decide whether content conforms to appropriate netiquette guidance. • Report posts or comments that violate community or legal guidelines.

Need more support? Contact us:

Call: 020 8203 1781 | Email: support@2simple.com | Visit: help.2simple.com

Powered by  2simple

Data Detectives

National Curriculum Links	Dominant objectives for this unit: • Select, use and combine a variety of software 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.
---	--

Declarative - By the end of the unit, the children will know that:	Procedural – By the end of the unit, the children will know how to:
• A database contains data organised in such a way that it can be queried to find useful information.	• Identify tables, records and fields. • Explain the data types contained within each field on a record. • Identify any format types applied to fields. • Use query tools to find useful information.
• Query tools exist to help users of a database find information from data stored within it.	• Use the filter tool to create conditions. • Use the grouping tool to group related information together as part of a query. • Use the calculate tool to apply calculations to selected data that provides meaningful results. • Use the sort tool to change the order records are presented according the field selected and the value order (increase/decrease).
• Chart tools can be used to make graphs from data contained within a database.	• Use the chart tool to select specific fields from a database for the x and y values on a graph. • Select the most appropriate graph type to display selected data so it makes sense in relation to a query given.
• Some databases will need multiple tables of data.	• Identify differences in data held in individual tables that make a complete database. • Explain why the data can't be all held on one table.
• Queries can be created that query data from multiple tables.	• Link tables together by using a common field. • Produce queries that query data from linked tables.
• Bespoke multiple conditions can be made that help a user find the exact information being asked.	• Create multiple condition and can join those using the AND/OR operator. • Use the AND/OR operator correctly and recognise the difference between the two and the impact this has on results when run.

	• Use brackets to contain multiple conditions and apply a general condition that applies to all conditions made.
--	--

Coding

National Curriculum Links	Dominant objectives for this unit: • 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.
----------------------------------	---

Declarative - By the end of the unit, the children will know that:	Procedural – By the end of the unit, the children will know how to:
• Cloning is a process where additional objects can be created while the program is running.	• Write code to clone objects and loop through the clones to change their attributes.
• Objects can be draggable in 2Code to provide another way for the user to interact with them.	• Make objects draggable in a program.
• Hotspots can be created in 2Code. These are a type of object that can be used to make a specific area of the screen react to actions such as objects entering or leaving the area.	• Create hotspots and write code that controls the flow of the program when objects enter and leave hotspots.
• Using a combination of coding structures and objects allows the design of interactive and creative programs.	• Use code to create effects when combining the use of draggable objects, hotspots and cloning in conjunction with sequences, repetition and selection.
• Variables can have different designated datatypes. The datatype will impact how the value of the variable can be used.	• Make conscious choices about datatype when creating variables. • Anticipate effects such as calculating using variable values, concatenation of string variables and bugs that could occur due to datatypes when user input is used.
• There are a variety of ways of receiving input from users and then using this input.	• Use 'Prompt for input' and 'get input' commands to receive input from the user of a program and write code using variables to work with user input.

	• Use if statements that include checking of a value of a user input and changing a variable if user input has met the if statement condition.
• A flowchart is a diagram that uses specifically shaped, labelled boxes and arrows to represent an algorithm as a diagram.	• Interpret flowcharts and use these as a basis to begin coding or debugging a program. • Create flowcharts to represent algorithms. • Include sub-routines within flowcharts to help with more complex program ideas.
• Using functions helps with making programs more efficient. Instead of writing the same sequence of code repeatedly	• Create functions and write code for function calls. • Interpret programs that include functions to explain the purpose of the function.
• Design documentation is vital when producing more complex programs. The process of designing supports abstraction and decomposition of the task at hand.	• Interpret design documentation, commenting on the level of abstraction and the way that a task has been decomposed. • Use user interface designs to create the look of a program. • Use flowcharts to decide upon the structure of a program and what functions might be required.
• Text-based adventures are computer games that have been created using text instead of graphics. Players use text commands to control characters and influence the environment.	• Review the code of a text-based adventure game and relate this to game play when the code is running. • Identify how variables, functions, selection and repeat loops are used to make the program function. • Adapt the code maintaining the structure but changing the game play.

Coding Crash Course

National Curriculum Links	Dominant objectives for this unit: • 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.
---	--

Declarative - By the end of the unit, the children will know that:	Procedural – By the end of the unit, the children will know how to:
• In computing, a set of instructions is known as an algorithm. Steps in an algorithm must be followed to achieve the intended outcome.	• Interpret what a piece of code means. • Follow a written algorithm. • Identify required events, objects and actions from a program algorithm. • Create instructions in the form of simple algorithms with attention to the order and the level of detail.
• Flowcharts are a type of diagram that use specifically shaped labelled boxes and arrows to represent an algorithm as a diagram.	• Identify features of a flowchart including the starts, inputs, outputs, delays such as a timer, actions and events. • Follow the flow of the chart and interpret what it is representing. • Use a flowchart as part of a planning process for designing a program.
• Programs follow a sequence of instructions (commands) in order.	• Run code and see the sequential effect that occurs.
• In 2Code, code is created using coloured code blocks that can connect to form commands.	• Make commands using the code blocks. • Run these commands.
• Commands consist of a combination of different block types; these are actions, events, objects and outputs	• Decide which type of block is needed and which combinations of block types will be needed to complete a command.
• An object is an item in a program that can be given instructions to change in some way by adding an action block.	• Use objects within commands when coding.
• An action is the way that an object changes when programmed to do so. For example, move.	• Connect object and action blocks to form commands.

Need more support? Contact us:

Call: 020 8203 1781 | Email: support@2simple.com | Visit: help.2simple.com

Powered by 

An event is something that makes a block of code run in response to an action such as a user pressing a key or clicking a screen.	- Use the 'When Clicked' and 'Collision Detection' event code blocks. - Position a command inside an event.
The way that code blocks are positioned within each other determines the order in which the code runs.	Create code that is nested for example putting an object and action command within an event command.
Selection in a program means that it has a decision command that will run different procedures dependent on whether a condition is met or not.	Interpret flowcharts depicting selection and explain what happens if a condition is or isn't met within it.
'If' statements and 'if/else' statements are used to create selection in 2Code.	- Identify 'if' and 'if/else' control blocks. - Read code that includes selection to determine what the program will do. - Create selection within 2Code using 'if' and 'if/else' statement blocks. - Debug programs that use selection. - Nest an if/else block inside an if/else block to give a user different validation options.
Co-ordinates are used in computer programming to determine the position of a point, shape or object and that these change according to where they are positioned on the screen.	- Identify co-ordinates in Design view. - Change the co-ordinates of objects within the code.
Repeat is a control block allowing blocks of commands to repeat a specified number of times.	- Identify situations when a repeat command can make the code more efficient. - Use a repeat command to make other commands repeat.
- Timers can control when a block of commands run. - Timer commands can be run after a timed delay or at regular intervals. - In 2Code there are two timer options, timer every or timer after. These can be altered by changing the number of seconds/quarter seconds.	- Code the features of timer after command including the interval, action(s), correct nesting of commands. - Distinguish between the use of the timer every and timer after commands. - Use a timer every command to make an event happen repeatedly.
- Variables are a place in computer memory that remains for the length of the program. - Variables have a name and a value. - The values can be changed in the code. - Variables are used in programming to keep track of things.	- Create number variables and initialise their values. - Change the value of variables within the program by using the 'Change Variable' command. - Use the value of variables within programs.

	Use the variable watch to monitor how a variable changes as the program executes code.
A datatype defines the type of data, for example, a number or text, and what operations can be performed with it.	- Make sensible choices of datatype when creating variables. - Anticipate the effect of operators on the variable values given their datatype. - Use the Variable Watch window to monitor the changing value of the variables in a program.
Concatenation is the name given to the action of linking things together in a series. For example, in programs we might want to link words together to form random phrases that are seen by a user of a program.	- Recognise where concatenation can be useful in programs. - Use the print to screen command in combination with random words that are joined together to demonstrate concatenation. - Demonstrate concatenation in other programs created such as linking a variable (score) with text (well done) for a game. - Debug programs that involve concatenation errors in strings.
- Testing and debugging are an important part of the process of making computer programs. - Understanding what nesting is and the effect it has on a program can help when trying to debug a program.	- Read the code of example programs with nesting to determine the outcome of running the code. - Test what happens when changing how a program is nested. Examples could include moving a timer so that it is nested within a timer that is currently nested within a when clicked command and comparing how it executes. - Use the knowledge of nesting to help debug a program that isn't working as intended.
Decomposition is a method of breaking down a task into manageable components. This makes coding easier as the components can then be coded separately and then brought back together in the program. Each part is known as a sub-routine.	- During planning, use decomposition to break down the plan into the key parts that are required to get the program functioning. - Use decomposition as a way to organise the code. For example, using tabs in 2Code. - Use 2Chart to create a flowchart that contains sub-routines.
A function is a named group of commands that a program can run when called by name; reducing the need to rewrite code repeatedly.	Create functions in 2Code and call these functions appropriately instead of repeating code.

- | | |
|--|---|
| | <ul style="list-style-type: none">• Refine existing code to include functions by recognising where code is repeated within the program. |
|--|---|

Introduction to Python

National Curriculum Links	Dominant objectives for this unit: • 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.
----------------------------------	---

Declarative - By the end of the unit, the children will know that:	Procedural – By the end of the unit, the children will know how to:
• Python is a text-based coding language.	• Use language syntax correctly for simple programs which includes the use of brackets, quotes, indents and correct case letters.
• Python has in-built functions that can be used.	• Use the print function to output strings, values of variables or a combination of both.
• A library of functions can be imported into a program to use.	• Import the sleep function from the time module within Python's library.
• A sleep function can be used to control how a program is executed.	• Use the sleep function to control the speed of execution of lines of code when printing outputs to the screen.
• There are two main types of loop in Python and that these can be used to improve coding efficiency.	• Use a 'for loop' to repeat a block of code for a specified time. • Explain the difference between a 'for loop' and a 'while loop'.
• Range is a function that lets a coder in Python get a series of numbers or specifies how many times a loop is to run.	• Use the range function in a 'for loop' to specify how many times that loop is to run. • Use the range function to start from a given number and end on a given number.
• Python can be used to perform mathematical calculations.	• Use mathematical operators with numbers to print answers to calculations to the output window.
• Sprites can be programmed to move to positions on the screen.	• Use design mode to select sprites to place on the screen. • Use coordinates to identify the position of a sprite. • Program a sprite to move to different places on a screen by using co-ordinates to identify current and intended location.

Need more support? Contact us:

Call: 020 8203 1781 | Email: support@2simple.com | Visit: help.2simple.com

Powered by 

Spreadsheets (Microsoft, Google and Apple)

National Curriculum Links	Dominant objectives for this unit: Select, use and combine a variety of software 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.
----------------------------------	---

Declarative - By the end of the unit, the children will know that:	Procedural – By the end of the unit, the children will know how to:
There are key features of a spreadsheet, and data can be entered into cells.	- Talk about some uses of a spreadsheet. - Navigate around the front screen of their chosen program. - Navigate around a sheet using appropriate cell references. - Enter data into cells of the chosen program. - Recall key vocabulary relating to spreadsheets.
Formulae can be entered into a spreadsheet, this can save time and make working more efficient.	- Define what is meant by a formula on a sheet. - Navigate around a sheet by clicking in a cell or typing cell reference. - Write simple formula related to the rules of calculation. - Use the series fill function. - Use formulae to change calculations automatically when data entered is changed.
Spreadsheet software can make complex data clear by manipulating the way it is presented.	- How to change the height and width of a cell and row. - Enter header titles into a sheet. - Split cells for ease. - Sort data in a sheet using the appropriate feature.
A spreadsheet can be used to model a situation.	- Use the sum feature when typing a formula. - Format cells as currency. - Use a computational model in a real-life situation.
Functions can be used for calculating averages and maximum and minimum values in spreadsheets.	- Add data to a table format. - Create formulae using cell references. - Use more advanced formula other than SUM
A spreadsheet program can display a variety of graphs and charts.	- Recap the knowledge of the different axis in a graph based on learning in other subjects. - Create a range of graphs and charts depending upon the version being used.

Need more support? Contact us:

Call: 020 8203 1781 | Email: support@2simple.com | Visit: help.2simple.com

Powered by  2simple

Using sorting and filtering tools can allow you to interrogate large amounts of data quickly and easily.	- Sort data in a sheet by alphabetical or numerical order. - Use and apply a variety of text and number filters. - Explain that rows that do not meet the criteria are not deleted; they are temporarily hidden from view. - Clear the filters after a search to make the hidden rows visible again.
A spreadsheet can be created to model a real-life situation and solve problems.	- Understand the advantages of using formulae when data is liable to change. - Plan an event using a spreadsheet. - Use the skills gathered over the unit to solve a series of problems.

3D Modelling

National Curriculum Links	Dominant objectives for this unit: Select, use and combine a variety of software 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.
----------------------------------	---

Declarative - By the end of the unit, the children will know that:	Procedural – By the end of the unit, the children will know how to:
Designers use CAD (Computer Aided Design) to create accurate drawings or 3D Models on a screen.	- Define what a 3D model is. - Identify different industries where CAD is used.
CAD allows designers to create accurate and flexible designs, allowing them to test out ideas and spot problems before something is built in real life.	- Identify the benefits of using CAD. - Explain how CAD is used in a variety of industries such as car manufacturing, architects and creating packaging.
2Design and Make is the CAD software on Purple Mash which allows users to develop a range of 3D models.	- Use 2Design and Make accurately, select models which fit their design best and adapt them where necessary. - Spot problems in their own 3D models on 2Design and Make and make adaptations where necessary. - Add textures to the model using the painting tools. - Use the design slider to alter the width of the model.
There are three viewpoints on 2Design and Make; Points view, Net view, and 3D view.	- View a model from each viewpoint. - Use each view to better their models, making use of the editing tools within each viewpoint.
The pattern fill tool allows you to design a repeated pattern to add to your model.	- Create repeated patterns that suit the model. - Add patterns into small sections of their model design.
Moving, adding and removing points on a model allows the designer to manipulate the shape to meet the desired design.	- Change the location of the points to alter the appearance of the model. - Add and remove points to manipulate the shape of their model to meet a purpose.
A prototype is a first version of a design which is designed and made to test out ideas.	Explain why a prototype is used in a range of industries before a real product is created.

Need more support? Contact us:

Call: 020 8203 1781 | Email: support@2simple.com | Visit: help.2simple.com

Powered by  2simple

• CAD can be used to create a prototype for packaging for a product.	• Explain how creating a packaging prototype is useful.
• Packaging is important to protect a product, give information, promote the product, attract customers and make the product more convenient to carry, use and store.	• Recognise a range of different packaging designs and shapes and think about how effective they are. • Discuss how packaging is used in a variety of ways.
• A design brief is a set of instructions or a challenge that explains what needs to be designed.	• Explain why design briefs are important. • Plan a design for a model to meet a design brief – a piece of packaging for a healthy snack.
• Models need to be refined before they are printed out using a standard printer or 3D printer.	• Refining a model is important prior to the final printing process. • Print their model onto paper/card or via a 3D printer. • Construct their 3D model if appropriate.
• A 3D printer can print a physical object from a 3D digital model, typically by laying down many thin layers of material in succession. It requires a STL file.	• Identify the differences between a standard printer and a 3D printer. • Identify a STL file. • Discuss the use of 3D printing in different industries. • Discuss the types of materials that 3D printers can use to create a physical object.

Binary

National Curriculum Links	Dominant objectives for this unit: • 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.
----------------------------------	---

Declarative - By the end of the unit, the children will know that:	Procedural – By the end of the unit, the children will know how to:
• Binary is a number system using only the integers 1 and 0. It is also known as base-2.	• Compare the integers used in decimal to those used in binary.
• In day-to-day life, we use the decimal system, also known as base-10 and denary.	• Compare the integers used in decimal to those used in binary.
• Binary is how data in a computer is saved and processed to make everything happen.	• Explain that the 0 and 1 integers in binary relate to the millions of switches within the computer being in an off state (0) or and on state(1). • Recognise when information is transferred and turned into action. For example, the letter 'A' appearing on the screen when the user types 'A' on the keyboard.
• A bit is the most basic unit of information in computing.	• Represent a single bit as a value of either a 0 or 1.
• The sequence of bits doubles each time e.g. 1,2,4,8,16,32	• Create decimal values by adding bits together.
• It is possible to count from 0 to infinity using binary.	• Count in binary using equipment such as binary cards to support them.
• All decimal numbers can be converted into binary.	• Convert decimal numbers to binary using the divide-by-two method.
• When coding, binary game states are often used to control the flow of the program, these values are stored using variables.	• Describe where coded programs might have used binary game states to control the flow. • Make a simple program that uses variable values of 0 and 1 to control the flow of the program.

micro:bits

National Curriculum Links	Dominant objectives for this unit: • 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.
----------------------------------	---

Declarative - By the end of the unit, the children will know that:	Procedural – By the end of the unit, the children will know how to:
• A micro:bit has pins that can be used to detect inputs.	• Connect crocodile clips to a numbered pin and the gnd pin to make an electrical circuit that the micro:bit detects as an input. • Use the 'when pin' command in Free Code micro:bit to create an event that triggers when an input with a pin is detected.
• Free Code micro:bit can be used in conjunction with the micro:bit's pins to create a range of outputs when a circuit is detected.	• Create variables that count the number of times a pin input has been detected and use this variable to display as an output. • Use text, image and sound outputs when an input is detected. • Use the sleep command to control when code is executed.
• A micro:bit has a range of sensors that can be used to collect environmental data.	• Identify the sensors on the micro:bit and explain how these could be used to collect useful information.
• 2Graph can be used to program a micro:bit to collect data.	• Connect a micro:bit to 2Graph. • Set up a micro:bit within 2Graph specifying the sensors to be used and the sampling data intervals. • Download the software onto the micro:bit and start data logging.
• Data collected by a micro:bit can be transferred into 2Graph and presented in a table and chart.	• Upload collected data from a micro:bit into 2Graph. • Modify data to be presented in the most readable way in order to analyse it.