Computing at Richmond Hill Primary Academy

Below are the Key Strands that our computing curriculum is designed around. Our computing curriculum supports children to develop Strands 1, 2 and 4 considering 3 prominent computing strands - computer science, information technology and digital literacy.

Strand 1- Developing Resilient & Aspirational Learners	Strand 2 – Developing Self-Regulated Learners	Strand 3 – Developing an Understanding of Equality, Diversity and Creating Culturally Rich Learners	Strand 4 – Developing Risk Assured Learners	Strand 5 – Developing Environmental and Sustainability Aware Learners

Ofsted's research review series: computing (2022) states, 'pupils' development of early computing knowledge is important. Grover, Pea and Cooper have suggested that: Learners' success in future engagement with computing will depend on how well introductory curricula prepare them in both the cognitive and affective dimensions of computational learning.'

How Our Computing Curriculum Has Been Designed

At Richmond Hill we teach Computing through the use of 'Purple Mash' Purple Mash offers a wide range of creative tools, games and resources for primary school pupils. It covers all components of computing and more, providing interactive activities and resources aligned with the National Curriculum. The platform aims to make learning fun and engaging while promoting digital literacy and critical thinking skills.

- **Computer Science:**

This strand focuses on the fundamental principles and concepts of computer science, including programming, algorithms, and data representation. It aims to develop computational thinking and problem-solving skills.

- **Information Technology:**

This strand focuses on the practical application of technology to solve problems and create content. It includes using software, manipulating data, and understanding how digital systems work.

- **Digital Literacy:**

This strand emphasizes the responsible and effective use of technology, including online safety, responsible digital citizenship, and critical evaluation of digital information.

These three strands work together to ensure that students are well-prepared to thrive in the digital world, both as individuals and as future professionals.

Purple Mash helps children to learn key computing skills including coding, data handling, animation, game creation as well as industry standard office software.

Ofsted's 2022 review draws a distinction between declarative and procedural knowledge in computing.[1] Declarative knowledge, often referred to as conceptual knowledge in the literature, consists of facts, rules and principles and the relationships between them It can be described as 'knowing that'. In contrast, procedural knowledge is knowledge of methods or processes that can be performed. It can be described as 'knowing how'. Purple Mash is designed with both procedural and declarative knowledge in mind, in a progressive and sequential way.

Each year group explores a sequence of units where each lesson within links to a following lesson in later year groups.

Rather than a scheme with set lessons, the early years resources are designed to integrate into the day-today routine and set-up of an early years setting with opportunities for using Mini Mash or Purple Mash as part of the Early Years curriculum to support children in working towards early learning goals.

Composite Knowledge

Key Stage 1

• In many units, children will be furthering online understanding and concepts of technology (DL) through making digital content (IT and CS)

Key Stage 2

• Children will be understanding the capabilities of the World Wide web (CS) while searching online (IT).

• They will be developing skills while learning about searching the Internet (IT).

Both Key Stages

At all times children will be learning about using technology safely and respectfully (DL).

In most units for all strands, children will be developing their general information technology skills (IT).

This overlap, repetition and reinforcement helps to give children a deeper understanding of the knowledge and skills across all strands and of their integrated nature in the real-world.

Our computing curriculum is supplemented by a stand-alone E-Safety curriculum as we identify this as a high need for our community. This curriculum is the 'common sense' education scheme. The Common Sense education scheme is aligned to the UKCISS framework and KCSIE 2024, as well as aligning to the PSHE/RSE Curriculum.

Pedagogical Strategies

Purple Mash embeds metacognition, formative feedback, balancing of cognitive load and spiral learning, which are evidenced as making a significant impact on raising attainment. Purple Mash motivates children to engage in independent, creative learning across the curriculum – building self-esteem as well as skills.

The pedagogical skills that we utilise through our computing lessons are revisiting prior learning, introducing new learning through modelling, use of discussion/talk partners moving into independent work. Childre experience lessons through the use of whole class teaching, with teacher modelling the use of appropriate technology, as well as the use of Chromebooks and IPads.

Year 1	Exploring Purple Mash, Grouping and Sorting 4 lessons	Pictograms, Lego builders 6 lessons	Maze Explorers 3 lessons	Animated Story Books 5 lessons	Coding 6 lessons	Technology outside of school 2 lessons
Year 2	Coding 6 lessons	Spreadsheets 6 lessons	Questioning 5 lessons	Effective searching 3 lessons	Creating Pictures 5 lessons	Making Music, Presenting Ideas 7 lessons
Year 3	Coding 6 lessons	Spreadsheets 6 lessons	Touch typing 4 lessons	Email 6 lessons	Branching Databases 4 lessons	Simulations and Graphing 4 lessons
Year 4	Coding 6 lessons		Writing for different audiences 5 lessons	Logo 4 lessons	Animation 3 lessons	Effective Search, Hardware investigators 5 lessons
Year 5	Coding 6 lessons	Spreadsheets 6 lessons	Databases 4 lessons	Game Creator 5 lessons	3D Modelling 4 lessons	Concept Maps, Word Processing (optional) 4 + 7 lessons
Year 6	Coding 6 lessons	Networks 3 lessons	Blogging 4 lessons	Text Adventures 5 lessons	Quizzing 6 lessons	Understanding Binary, Spreadsheets 12 lessons

Unit 1.2 – Grouping & Sorting

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Computer Science** • Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Items can be sorted using a range of criteria. When sorting items, a logical process should be used.	• Describe physical items that are needing to be sorted thinking about all the different ways they could be described. • Identify criteria that can be used to sort items into two groups. • Explain how items have been sorted. • Check that items sorted into two categories are correct using the criteria decided upon.
• An algorithm is a precise, step-by-step set of instructions used to solve a problem or achieve an objective. Computer programs need clear instructions, in steps, to follow. The instructions written for a computer program are called algorithms. Humans can follow algorithms to sort items such as shapes, just as computer programs can.	• Look at an algorithm a human has followed to sort shapes and compare it to the algorithm a computer program has used to identify if the shapes are correctly sorted. • Follow a human algorithm to sorting shapes. • Follow a computer program algorithm checking shapes have been sorted correctly.

• Computers can be used as a way of sorting on screen objects.	• Open a sorting activity within Purple Mash. • Become familiar with the layout of computer sorting activities recognising items that need sorting and the areas they can be moved to. • Identify what each criterion container is. • Drag objects into the correct criterion container. • Recognise some objects may fit into an overlap criterion.

Unit 1.3 – Pictograms

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • Use technology purposefully to create, organise, store, manipulate and retrieve digital content. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Data is a collection of information, used to help answer questions.	• Collect data on a common theme such as how children travel to school. • When collecting data, recognise that there are efficient ways of collecting data such as writing it down or entering it into a computer program.
• A pictogram is a visual way of representing data.	• Represent data collected as a class using physically created pictograms. • Interpret a pictogram by comparing amounts of different categories.
• We can look at data represented in pictograms and ask questions as a way of interrogating data.	• Interrogate a pictogram by thinking of questions that we would like answers to. • Look at a pictogram and compare each category. • Identify the totals in each category.
• Programs such as 2Count enable people to create pictograms on a computer. This has the advantage of being able to easily modify data and share it with lots of people.	• Open 2Count. • Increase or decrease amounts of items from a column by using the plus or minus buttons. • Change an image representing a piece of data. • Create a suitable title for a pictogram. • Save a pictogram.

Unit 1.4 – Lego Builders

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Computer Science** • Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• To achieve a specific effect when building something, accurate instructions must be followed.	• Recognise whether instructions have been followed correctly when comparing two Lego models. • Give clear, precise and concise building instructions for someone to follow. • Recognise how important it is to have clear, precise and concise instructions and the implications of this. • Test that instructions have been followed by comparing the results of something built with the instructions.
• Computer programs need precise instructions to follow, and these are called algorithms. If instructions are vague, outcomes will vary for any given task.	• Open a painting activity on Purple Mash. • Follow the simple instruction of painting given animals and compare the finished results with others recognises differences due to limited instructions given. • Follow a set list of instructions that everyone uses to paint a bird, recognising that the instructions have resulted in everyone's finished pieces are very similar.
• The order of instructions for a task affects the results.	• Identify why a sequence of instructions for making a sandwich is incorrect.

• Correcting errors in an algorithm or program is called debugging.	• Find simple errors in a simple algorithm for making a sandwich. • Correct the algorithm sequence by re-ordering it. • Recognise when an algorithm has been debugged. • Apply learning about debugging an algorithm to other incorrectly sequenced instructions such as baking cakes.

Unit 1.5 – Maze Explorers

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Computer Science** • Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • Create and debug simple programs. • Use logical reasoning to predict the behaviour of simple programs. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• You can move a character (turtle) within specific computer programs around a computer screen such as 2Go by using direction keys. When a direction key is used it is known as a command.	• Open 2Go and be familiar with its environment. • Use the direction keys to make a character (turtle) on the screen move in different directions. • Experiment with moving the character using alternative routes to get it to a desired location.
• On screen direction keys can have eight possible directions which includes diagonal movements.	• Make use of diagonal key commands when moving a character to help move the character to a desired location with the least number of commands. • Combine diagonal commands with standard four direction commands and number keys to efficiently move a character to a desired location.
• Number keys can be combined with direction keys to give a program more accurate instructions and avoid less command clicks. Each square on a grid relates to 1 unit and that when using number keys this should be referenced.	• In 2Go use the direction keys combined with number keys to get an object to a specific place on a screen. • Reference an onscreen grid with number keys when creating commands.

• Lists can be made with directional instructions within 2Go and these are known as algorithms. These lists can be changed to improve the instructions which is known as debugging.	• Identify where a character needs to go. • Formulate a list of instructions to move the character from the start to end point. • Drag instructions into the algorithm box. • Run the instructions and test they achieve the correct result. • Debug by modifying the instructions so that the character moves to the correct location. • Make use of the undo button to help with changes to commands. • Use the extend algorithm button when more than five commands are needed.

Unit 1.6 – Animated Stories

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • Use technology purposefully to create, organise, store, manipulate and retrieve digital content. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• There are differences between traditional books and e-books.	• Identify differences and similarities between traditional books and e-books. • Explain the advantages of both formats of books and why one format might be favoured over the other in some instances. • Identify 2Create a Story as an e-book creator tool.
• Images can be created within e-book software.	• Open 2Create a Story and explain what the common tools such as eraser, undo/redo do and the textured pens. • Use the textured pens to create a drawing. • Use the eraser, undo/redo buttons when creating the image if a mistake is made. • Save the 2Create a Story file.
• Animations can be included in e-books.	• Open previously saved work from within 2Create a Story. • Identify the animation tool. • Test each animation effect within the animation tool for a selected image. • Apply an animation effect. • Use the play button to see the effect of the animation within the e-book.

• E-book software allows pages to be added and overwriting of work.	• Overwrite any work that was done previously if needed such as changing an image. • Add additional pages using the add page button. • Scroll between pages using the back and forward arrows. • Save any new changes.
• Audio such as sound effects, voice recordings and music can be included within e-books.	• Open previously saved work from within 2Create a Story. • Locate the sound button. • Record sound using the microphone and apply to a page. • Insert a sound effect from the gallery and apply to a page. • Insert a piece of music created from the piano synthesizer and apply to a page. • Test the effects of adding sound by clicking the play button.
• Backgrounds can be included in e-books to help engage an audience.	• Open previously saved work from within 2Create a Story. • Locate the clip art gallery icon. • Select a background for a page from the gallery. • Create a background for a page using the pen tools. • Locate the camera icon and use this to apply an image taken from a camera as a background to a page.
• Text fonts and sizes can be changed in e-books to suit an intended audience.	• Select previously written text from a file. • Locate text button. • Experiment with changing the font type, colour and size. • Apply any text changes to a page that contains text.

• Copy and paste features in e-book software can be used to speed up creation of additional pages.	• Open previous work within 2Create a Story. • Locate the copy button. • Select copy to perform a copy of a page. • Use the next page button to locate where the copied page should be pasted. • Locate the paste button. • Click on the paste button to insert the copied page. • Modify the copied page.

Unit 1.7 – Coding

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Computer Science** • Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • Create and debug simple programs. • Use logical reasoning to predict the behaviour of simple programs. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Tasks can be given to people and computers by using instructions. Computer programs work by following instructions called code known as algorithms. In both cases, these need to be clear and concise.	• Give clear instructions that others can follow. • Receive instructions that others can follow. • Draw symbols to represent instructions. • Can recognise an object in printed code block form. • Can recognise an action in printed code block form. • Can arrange a printed object block next to a printed action block e.g. 'Tuna left'. • Can recognise the set of code blocks arranged to create actions is known as an algorithm.
• There are objects and action code block in the 2Code environment and that you can make a simple program using these. Each single instruction such as 'Object Right' is called a command.	• Recognise object code blocks in 2Code. • Recognise action code blocks in 2Code. • Make a command in 2Code by using an object and action together. • See what happens when a command they have made is executed.

• An event is something that makes a block of code run such as a user pressing a key or clicking a screen. Event, object and action code blocks can be used together.	• Recognise **When Clicked** code block as an event block. • Arrange a **When Clicked** code block in front of an object. • Give an object code block an action when it is clicked. • Run code with a **When Clicked** event and observe what happens when the event occurs.
• When code is run this is known as code being executed.	• Execute code by clicking the Run button. • Stop code executing by clicking the Stop button. • See the colour change on blocks of code being executed.
• Debugging is when we fix code that isn't working how it was designed to.	• Analyse where their code isn't working properly. • Arrange blocks into different places. • Change actions attributed to objects. • Can execute code and test if changes have debugged a simple program.
• Scenes can be made using backgrounds and objects. Backgrounds can be changed as well as objects and that these have attributes (properties) that can be modified.	• Switch to design view. • Select a background using the background icon. • Click on an object and change the size of it by changing the value of the scale. • Move an object where wanted within design view by clicking and dragging it. • Delete an object by clicking on it and then on the bin.
• A well thought out program should be made from a plan.	• Draw a plan of a scene with objects. • Plan what the objects in the scene will do. • Create a program from a plan that includes objects, actions and a **When Clicked** event. • Execute the program and test if it is doing what is intended in the plan. • Debug the program if the program isn't working how it was planned.

Unit 1.9 – Tech Outside School

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Digital Literacy** • Recognise common uses of information technology beyond school. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Technology is science and engineering knowledge put into practical use to solve problems or invent useful tools.	• Recognise technology. • Identify common types of technology such as electronic devices. • Describe the function of common types of technology.
• Technology is used within school.	• Identify technology within school. • Describe the function of technology examples within school. • Explain how this technology is helpful.
• Technology is used outside of school.	• Identify technology outside school. • Describe the function of technology examples outside school. • Explain how this technology is helpful.

Unit 2.1 – Coding

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Computer Science** • Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. • Create and debug simple programs. • Use logical reasoning to predict the behaviour of simple programs. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• In computing, a set of instructions is known as an algorithm. Steps in an algorithm must be followed in order to achieve the intended outcome.	• Follow a written algorithm on a plan and interpret it. • Identify events, objects and actions in a plan of an algorithm. • Predict what will happen if the planned algorithm was converted to code. • Use the correct code within 2Code to implement the algorithm which includes event 'when clicked', objects and actions.
• Code can be created that detects when two objects have collided. This code can have an action associated with it. For example, if an alien ship (object) collides with a planet (object) a crash sound is heard (Action). We call this collision detection in 2Code.	• Recognise the **collision detection** block as part of the event category blocks. • Drag a **collision detection** block into a program. • Assign two objects within the **collision detection** command for it to detect collisions. • Assign an event for when the two objects collide such as when princess collides with frog, princess says 'hello'.

• Programs follow a sequence of instructions (commands) in order. Timers can be introduced into programs to make parts of the program run after a set time. In 2Code, you can use a timer after command to delay the number of seconds until specific parts of a program are run.	• Recognise the **timer** block as part of the control category blocks. • Drag a **timer** command block into a program. • Use the **time after** command and set a number of seconds. • Place code within a **timer after** command that will run once the timer has reached the set seconds after execution. • Observe placing two separate **timer** commands in a program and identify that they will run independently of one another if a timer isn't nested inside another timer.
• A computer program in 2Code can include objects that are different types. Each object type will have attributes (properties) that can be modified.	• Go into design mode of 2Code and find the different objects. • Place up to four different objects into a design scene of a program including the 'Turtle' object. • Change an object image by clicking on an object and selecting image from the attributes table. • Change an objects size by clicking on an object and selecting scale from its attributes table.
• Events in computer programs cause a block of code to be run. Events could be the result of a user pressing a key or clicking the screen. Event commands in 2Code are used to create blocks of code that are run when an event happens. There are different event command blocks in 2Code.	• Recognise the event command blocks – **When Key Event**, **When Swiped Event**, **When Clicked Event** and **Collision Detection**. • Include a **When Key Event** or **When Clicked Event** in a program. • Assign up to four objects to the **When Key Event** or **When Clicked Event** that have been created in design mode. • Give each object a movement. • Run code and test that when the event occurs e.g. 'click up arrow', all four objects move.

• Buttons are an object type in 2Code. Buttons use the 'When Clicked' event and will run a piece of code when they are clicked on.	• Enter design mode and locate the button object under 'Form/Text' tab.
	• Insert a button into design mode scene that contains other object types. • Drag the button command object when in code view into a program and notice that it contains the **When Clicked Event**. • Nest code within the **When Clicked Button** that makes an object carry out an action when the button is clicked.
• Bugs when referring to computer programs, are bits of code that are stopping a program from working how it was intended. Debugging is the process of looking for any problems in code, fixing the problems and repeatedly testing them.	• Recognise what a program in 2Code is supposed to do. • Run the code and check that the program is operating correctly. • If anything hasn't worked correctly, the code responsible for the area is located. • Stop the program and make changes to the broken code. • Run the program again and test it. • Repeat the process until the program is running as intended.

Unit 2.3 – Spreadsheets

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • Use technology purposefully to create, organise, store, manipulate and retrieve digital content. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• There are specific features and purposes of a spreadsheet, and they can navigate around and enter data.	• Talk about the function of a spreadsheet and give examples. • Enter numbers and words into a sheet. • Navigate from cell to cell using the arrows or by clicking in the cell. • Add background colour to cells. • Explain what a row and column is in spreadsheets. • Add a given image to a sheet or draw their own. • Allocate a value to an image and then recap the use of the count tool.
• Specific features in spreadsheets such as 2Calculate allow user to insert content such as images into a cell. The cells content can be locked or moved using additional features.	• Select a cell where an image will be inserted. • Locate the clipart library and add an appropriate image to a single cell. • Select multiple cells to add repeated versions of the same image. • Use the Move tool to move an image from one cell to another.
• The totalling tool counts all the cells behind the tool.	• Know what is meant by totalling numbers in a row or column. • Know how to use the totalling tool. • Practice using the totalling tool.

	• Practically apply the knowledge to make a magic number quare where the totalling tool automatically adds up rows and columns.
• Spreadsheet can be used for calculations.	• Use the + and – to create simple formulas to calculate an amount. • Work out if the coins presented in the sheet equal or don't equal a given amount.
• Data in a spreadsheet table can be edited and used to create a block graph.	• Add a title to their chart, • Label the columns in a table and enter data into it. • Use the table data and graphing tool to create a block graph.

Unit 2.4 – Questioning

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • Use technology purposefully to create, organise, store, manipulate and retrieve digital content. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Pictograms created through software or physically are of limited use beyond answering simple questions.	• Create a class pictogram using 2Count. • Identify questions we can ask and find the information for on the pictogram. • Identify what questions we can't ask due to a pictogram being limited with the information it can provide.
• Information can be separated by using yes/no questions.	• Create suitable yes/no questions for a collection of physical data such as avatars of children on individual paper records. • Can use yes/no questions to find individual paper records. • Can recognise that not all yes/no questions will work and you are limited with the information on individual records.
• A binary tree is a simple way of sorting information into two categories. When using a binary tree, users can only ask yes/no questions to find a specific piece of information.	• Design a binary tree physically using paper to sort simple pieces of information such as animals or children. • Use a pre-populated binary tree program such as 2Investigate to find answers to yes/no questions.
• Databases are a computerised system that make it easy to search, select and store information. Databases contain records which have a variety of information about a specific entry.	• Open a 2Investigate database. • Identify the records which make up a database. • Click on individual records. • Identify fields as pieces of information collected for a record.

• Users can search a database using simple and more complex search questions.	Reference given questions to find specific records that meet a search query by: • Locating the find tool. • Using the drop-down lists for record fields in the search tool. • Combine more than one identifier such as ‘has glasses’ and ‘brown hair’.

Unit 2.5 – Effective Searching

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Digital Literacy** • Recognise common uses of information technology beyond school. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• The Internet is a global network of connected computers around the World.	• Answer questions accurately about what the Internet is by completing a quiz. • Explain the difference between the Internet and the World Wide Web, recognising that the World Wide Web is powered by the Internet.
• The World Wide Web refers to the documents and pages someone sees when using a browser. Websites can be found using a browser that contains a search engine.	• Recognise a web browser. • Recognise a search engine and the key elements they contain. • With guidance, enter a search query in a search engine. • Review results from a search query. • Find the number of results for a query entered into a search engine. • With guidance, use some of the search tools on a search engine such as: all, images and news.
• Search engines use millions of people's digital footprints to help provide more accurate results.	• Review a search results page. • Discuss with others that a digital footprint is a record of individuals interactions online and that this is used to help search engines provide better results for individuals.

• To find results that we want on a search engine, we need to search effectively.	• Search using words. • Search using questions. • Compare using words to questions in a search engines results page. • Share information about searching effectively by creating a leaflet for others.

Unit 2.6 – Creating Pictures

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • Use technology purposefully to create, organise, store, manipulate and retrieve digital content. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Computer drawing programs contain palettes. Palettes are the range of colours or shapes available to the users.	• Open 2Paint a Picture. • Select a painting effect. • Use the colour palette to select different colours.
• Computer drawing programs may have a choice of painting effects. Painting effects can be combined to help a user make pictures.	• Explore the range of painting effects on offer in 2Paint. • Observe how the painting effects give different results. • Produce a range of paintings formed from different painting effects.
• The size of an onscreen painting tool brush stroke can be manipulated.	• Select different colours for a painting tool. • Locate the changing scale brush tool slider. • Use the brush tool slider to change the size of brush strokes.
• Intensity of colours can be manipulated.	• Identify the dilute tool. • Select a colour from the palette. • Experiment using the dilute tool to manipulate the intensity of any selected colour.

• Outline features in drawing programs help a user with the formation of paintings.	• Identify the outline tool. • Select an outline of choice. • Resize an outline using the draggable blue squares. • Change the points on an outline by using the draggable green squares. • Paint within the outline. • Position the outline on the page where needed.
• Fill tools speed up the process of colouring enclosed areas on a painting.	• Locate the fill tool. • Use the fill tool to place colour into enclosed areas on a drawing.
• Pattern tools can be used to create repeating patterns and manipulate how a pattern is arranged.	• Locate the pattern template. • Draw within the square. • Change the arrangement of the pattern using the arrangement choice options. • Increase or decrease the size of the pattern and how often it is repeated by using the pattern slider.
• An eCollage template is available in programs like 2Paint that let's a user create stamps that can be used to add to a picture or build up a picture.	• Select the eCollage template. • Draw within the square. • Combine drawing by using the clipart library.

Unit 2.7 – Making Music

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • Use technology purposefully to create, organise, store, manipulate and retrieve digital content. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Music can be made digitally using programs like 2Sequence.	• Open 2Sequence. • Listen to a premade composition by pressing the play button. • Observe what happens on the screen when composition is playing including the placement of sounds.
• Sounds can be incorporated into music programs to make a melody.	• Open 2Sequence. • Explore the sounds and instruments category. • Drag sounds into the playable area. • Play and experiment with the organisation of sounds on each track.
• The speed of a digital musical composition known as tempo can be altered.	• Locate the beats per minute slider. • Experiment with changing the slider's position. • Listen in play mode to how changing the position of the slider affects a composition.
• The volume of instruments/sounds on a track can be changed when using music programs.	• Locate each bar in a composition. • Locate the volume sliders for each bar. • Adjust the volume on individual bars. • Play the composition and test the effect of altering the volume of individual tracks.

• Additional features, such as changing the number of bars and looping a composition, are available in music programs.	• Locate the bar selector. • Change the number of bars for a composition and observe how the composition changes when played. • Experiment with the looping feature. • Identify how this affects a composition. • Manipulate a composition so that it sounds correct when looping and doesn't have a noticeable jump when it restarts each repeat.
• Music programs let users incorporate their own sounds into a composition.	• Locate the My Sounds section. • Click on the record icon. • Use the record button to incorporate own sound from a microphone. • Test the recording back by pressing play. • Click done when happy with recording.

Unit 2.8 – Presenting Ideas

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • Use technology purposefully to create, organise, store, manipulate and retrieve digital content. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Digital content can be presented in many forms.	• Compare a traditional book with an e-book and can talk about the differences. • Recognise digital concept maps and their use for organising ideas. • Discuss the differences between a traditional book, e-book, concept map and digital quiz including the advantages and limitations of each format.
• Quizzes can be made using programs such as 2Quiz.	• Open 2Quiz. • Explore the front screen of 2Quiz and identify the key areas such as introductory screen, delete, clone, add questions, preview and play quiz. • Add a question type to 2Quiz. • Recognise some of the differences between question types.
• Digital content should be presented using a suitable format	• Compare a digital mind map in 2Connect with a digital fact file in 2Publish. • Discuss and explain the two formats. • Identify the format that is of most use when presenting to an audience.

• Digital content in one format can be re-used in other formats to present to audiences	• Open a 2Connect file with information on it. • Open a 2Publish file. • Use the 2Connect file to support creating content in the 2Publish file. • Use font tools, clipart, page settings and images to enhance digital content in the digital publishing file.

Unit 3.1 – Coding

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Computer Science** • Use technology purposefully to create, organise, store, manipulate and retrieve digital content. • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Flowcharts are a type of diagram that use specifically shaped labelled boxes and arrows to represent an algorithm as a diagram.	• Identify the point the flowchart starts. • Identify any points on it that represent an input or output. • Identify any delays such as a timer. • Identify any processes. • Follow the flow of the chart and interpret what it is representing. • Create a representation of the flowchart by using 2Code.

• Timers are used in coding to help control when a block of commands are 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.	• Review use of **timer after** command from previous year. • Insert a **timer after** command in code view and specify number of seconds. • Insert code within the timer that will action after specified seconds. • Nest a second timer within a **timer after** command knowing that the second (nested timer) will run only after the first timer has finished. • Begin to distinguish the difference between **timer every** command from the **timer after** command. • Use a **timer every** command to make an event happen such as a ticking sound for a clock every second.
• **Repeat** is a control block and blocks of commands can be set to repeat a specified number of times using the **repeat** control block.	• Understand that the repeat command is useful for avoiding lots of unnecessary coding repetition such as when a screen turtle is used to draw a square. • Identify the **repeat** command as part of the control blocks group. • Insert a **repeat** command into the coding area and set it a specified number of times to repeat. • Add a block of commands to a **repeat** command. • Execute the code and check that it has operated as intended.

• Testing, debugging and fixing 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.	• Recognise examples of nesting in a 2Code program. • Compare two example programs that both have nesting and are trying to achieve the same outcome. • Identify what happens if a timer is nested inside a **when clicked** event command and know that the timer will only initiate after the **when clicked** command has run. • 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.

Unit 3.3 – Spreadsheets

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Graphs can be generated from data within a sheet. If data is changed on the sheet, then the graph automatically updates to recognise these amendments.	• Recall the different range of graphs and charts they have come across in other subjects as well as computing including pie and bar. • Enter data into a table format in a spreadsheet. • Select all the data in the table. • Select the chart tool. • Give the table a title. • Label the chart axis. • Add a title to the chart. • Edit data in a table and see how the chart changes automatically. • Interpret the data contained within the graph including estimating values between given data sets.
• Cells all have their own individual address. They are referenced using letters and numbers.	• Read a cell address using column: row cell address. • Click in a given cell by using the cell address. • Complete a task to show their knowledge of cell addresses.

• Formulas can be added to a spreadsheet to speed up calculations when data is changed.	• Understand what a formula is in a spreadsheet. • Find the formula wizard box in 2Calculate.
	• Follow the steps of the formula wizard to create formulae using the 4 operators. • Write formulae directly in the formula bar knowing the need for the equals symbol to denote a formula
• There is specific functionality of some of the tools within 2Calculate.	• Locate the 2Calculate tools on the main screen. • Use the random number tool. • Use the spin number tool. • Use the timer tool. • Apply this knowledge to a specific task.
• Spreadsheets can be used to model a real-life situation and improve the efficiency of day-to-day tasks.	• Recall what is meant by a budget and why budgeting is important. • Analyse the information in a budget setting spreadsheet. • Create their own budget template on 2Calculate. • Use the TOTAL functionality within the spreadsheet program. • Make amendments to a budget spreadsheet and see what impact that has on the totals.

Unit 3.4 – Touch Typing

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Typing is the action or skill of writing something by means of a keyboard (physical or virtual) and that it is important to have a good posture when typing.	• Check that posture is correct when typing. • Position equipment correctly so that eyes are level with monitor. • Place feet on the floor. • Position wrists so that they are not touching anything when typing.
• Home, top and bottom row keys are areas on a keyboard where specific keys are located.	• Locate the home, top and bottom keys. • Open activities in 2Type referencing the keyboard guide to support recognition of using the correct key when typing letters.
• To be an efficient at typing hands should be positioned correctly on a keyboard and that the left and right hands should work independently of each other.	• Use the left hand to type letters. • Use the right hand to type letters. • Position the left and right hands correctly. • Build up to combing left and right-hand use to type words.

Unit 3.5 – Email

NATIONAL CURRICULUM LINKS	**Dominant strands for this unit: Digital Literacy & Computer Science** • Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact. • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• There are different methods of communication and they each have strengths and weaknesses.	• Present different methods of communication on mind mapping tool. • Explain the advantages and disadvantages of each method. • Discuss the development of newer communication methods and how they came to being.
• Emails are electronic versions of letters, and they can be sent and received almost instantly to anyone with an email address.	• Open 2Email. • Identify key areas and functions: Inbox, alerts, reply, formatting tools. • Open an email and reply to it. • Compose an email including address, subject and message.

• It's important to use email systems safely and that there are things people can do to try to keep themselves safe.	• Recognise a concerning email/contact. • Report a concern to a teacher verbally as well as the report to teacher feature in 2Email. • Discuss the disadvantages of email in regard to safety. • Identify what a trusted contact is. • Limit the information shared using email. • Recognise personal and private information and how to distinguish between them. • Use the draft feature to review messages before sending them.
• Pictures, documents and other file types can be attached to emails.	• Identify the attachment icon. • Select files to attach to an email and send. • Be cautious of email received that have an attachment. • Discuss the advantages and disadvantages of being able to send attachments with emails.
• Address books can be made in email clients which store known contacts' email addresses. When sending an email we can use an address and send to multiple people.	• Use the address book within 2Email to find contacts. • Send an email to multiple contacts using the address book. • Use carbon copy correctly and explain scenarios of when this might be useful. • Use blind carbon copy and discuss scenarios of when this might be useful.

Unit 3.6 – Branching Databases

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• A database is a collection of data organised in a way that it can be searched, and information found easily.	• Explain what a database is. • Provide examples of common uses of a database such as the school's attendance database.
• Objects can be sorted using yes/no questions and relate this to how computer binary databases work.	• Explain binary databases are also known as branching databases due to the branch like structure. • Identify questions that can be used to sort physical objects. • When a question is asked their can only be two possible answers. • Sort physical objects using appropriate yes/no questions. • Develop questioning to include more/less.
• Branching databases can be created using programs such as 2Question.	• Locate and open 2Question. • Add record cards within 2Question using a plan. • Insert question texts and choice button texts for each card. • Include an image for each card. • Use the final answer card option for end of a branch.

• It is important to test and debug if needed when creating branching databases so that they work as intended.	• Plan a branching database. • Use 2Question to create own branching database. • Work through all routes on the database and test whether it works as intended. • Identify errors. • Fix errors and test again.

Unit 3.7 – Simulation

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Computer simulations are programs that model real-life situations. They allow people to test various scenarios out that might be too expensive or dangerous to do in real life.	• Explain what computer simulations are and why they are useful. • Provide examples of computer simulations. • Give positives and negatives for simulations.
• Computer simulations can be realistic and also unrealistic depending on how well thought out they are.	• Explore a simulation in 2Simulate. • Make decision based on the options the simulation has given. • Find solutions to problems encountered when exploring a simulation.
• It is important to analyse and evaluate simulations to assess their usefulness and overall realism.	• Explore a simulation in 2Simulate. • Evaluate a simulation to determine its usefulness for purpose. • Evaluate the realism of a simulation.
• Simple simulations can be created using familiar software such as 2Create a Story.	• Plan and create a simple simulation in 2Create a Story. • Consider the relationships and rules in which the simulation uses.

Unit 3.8 – Graphing

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Computer programs such as 2Graph can be used to present data in more meaningful ways.	• Open 2Graph file with prepopulated data. • Compare how the data is presented in the table and the graph. • Edit an existing graph to show updated information that has been recorded.
• It's important to use the most appropriate graph type according to the information entered into it.	• Recognise the different types of charts that can be used to display data recorded within 2Graph. • Compare using different charts for data entered into 2Graph. • Identify the most suitable graph to display the data. • Explain why some charts are more suitable than others for displaying the data.
• Graphing programs can be used to help solve questions.	• Investigate a topic such as the number of times a particular number lands when a dice is rolled x times. • Collect data in a suitable table. • Record the collected data into 2Graph. • Include accurate labels and a title. • Answer questions from the investigation using the graph created.

Unit 4.1 – Coding

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Computer Science** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• There are objects in 2Code and that there are different types and these have attributes (properties) that can be changed.	• Enter design mode and locate the game elements tab. • Distinguish between the object types. • Click on an object and identify the attributes (properties) that can be changed. • Change attributes such as image, name, allow off screen, angle and movement. • Drag and position objects around the design mode screen. • Include objects of choice within the design screen.
• Backgrounds can be changed and manipulated.	• Enter design mode and locate the background icon. • Select a background of choice by clicking on the image attribute. • Include background of choice within a program they are creating.

• Selection is a term used in computer programming. That it is a decision command that will be run dependent on whether a condition is met.	• Recognise flowcharts and what each node means. • Use a flowchart to help them visualise a simple program. • Interpret flowcharts depicting selection and explain what happens if a condition is or isn't met within it.
• **If** statements are used to create selection in 2Code and that they are bits of code that will run only if a condition is true.	• Identify **if** Statement control blocks. • Recognise how an **if** Statement in 2Code is being used to create selection within a simple program. • Create selection within 2Code using **if** statement blocks within own program.
• Coordinates 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.	• Open design view. • Click on object(s) to expose attributes. • Identify where coordinates x and y can be changed. • Change the coordinates in the attributes of objects.
• **Repeat until** is a control block and that blocks of code will repeat until a condition is met.	• Identify r**epeat until** within the control blocks of block code view. • Look at code with r**epeat until** and know how to change code within it to meet an expected outcome. • Insert **repeat until** into own programs. • Successfully create code within 'Repeat until' block and make it run until a condition is met.
• **If/else** statements are a conditional command that tests a statement. If a condition is true, commands inside the **if** block will run. If a condition is false, commands inside the **else** block will run.	• Identify the **if/else** block within the control blocks of block code view. • Insert the **if/else** command within a program. • Create an **if** statement using blocks of code. • Create an **else** statement using blocks of code. • Execute the code checking the **if/else** code runs as expected.

• Variables are a virtual container (A place in computer memory) that contain a value that can change. The value is normally in the format of a number or letter. Variables are used in programming to keep track of things that can change such as the score in a computer game. There are 3 main types that be created using 2Code.	• Identify the **create variable** command from the variables blocks within code view. • Drag a **create variable** block into coding area. • Set a variable type to number. • Name a variable. • Set the value of a variable. • Execute code with a variable within it. • Use the variable watch to monitor how a variable changes as the program executes code.

Unit 4.4 – Writing for different audiences.

NATIONAL CURRICULUM LINKS.	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Formatting including the style of font can affect the impact of a text.	• Look at the different fonts used in a range of texts. • Look at the way a font has been used for effect. • Look at examples of unformatted text and how difficult it is for the reader to understand.
• Editing the formatting of the text makes a document fit for purpose.	• Look at the formatting options available to users on the formatting toolbar. • Change the style of font. • Change the font alignment. • Change the colour of the font.
• Producing documents to meet a brief involves using appropriate formatting.	• Make notes about the event. • Type up the event as a newspaper report using appropriate formatting choices. • Produce a persuasive document using appropriate formatting choices.

Unit 4.5 – Logo

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Computer Science** • Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. • Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• 2Logo has its own language with specific instructions.	• Input directional instructions into 2Logo and leave appropriate spaces. • Input the more abstract non directional code such as PU, PD and CS.
• Representations of shapes, letters and flowers can be created in 2Logo using the repeat command.	• Follow simple instructions to create simple shapes initially on paper and then on the screen. • Create their own shapes by typing all the instructions and then using the repeat command. • Follow instructions to create letter shapes and four-letter words. • Create 2Logo flowers.
• The repeat command is a more efficient way to code in 2Logo.	• Program repeating commands of code a line at a time. • Use the repeat command to replicate the same outcomes as repeated typing of commands. • Create regular shapes using the repeat command in preference of writing long code blocks.

• It is important to test and debug code in 2Logo as with other coding platforms to ensure it runs effectively.	• Look at the screen and isolate errors in the output. • Look at the code and identify why the errors have occurred.

Unit 4.6 – Animation

NATIONAL CURRICULUM LINKS.	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Some animations are created by hand and others with the help of technology.	• Talk about what is meant by hand made animation. • Make a simple flick animation book. • Open 2Animate on Purple Mash and discuss why animation using technology may be easier than using hand drawn images. • Create a simple moving object animation on Purple Mash using 2Animate.
• Onion skinning is a term used in animation and can make the animation process more efficient.	• Appreciate the purpose of onion skinning. • Use the onion skinning tool in 2Animate and they can talk about how it speeds the process up.
• Sound can be added to animation to enhance the finished product.	• Add sounds and background to their animations to improve them.
• The term stop frame animation refers to animation where the stopping and starting of a camera gives an object the impression of movement.	• Talk about stop frame animation they have watched, and the techniques used. • Create a simple stop frame animation. • Present their animation to an audience.

Unit 4.7 – Effective Searching

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Digital Literacy** • Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Information can be located on a search engine page.	• Load up a search engine onto their device and give the name of a well-known search engine. • Enter the search enquiry. • Research the different types of information one can get from a search engine. • Correctly interpret the information outputted.
• There are different skills needed to research effectively.	• Enter basic search enquiries. • Enter more advanced effective enquiries without the need for full sentences. • Answer a quiz using effective search.
• Web Pages need to be evaluated to see if the information contained is true and reliable.	• Analyse the contents of a web page for clues about the reliability of information. • Appreciate that the search engine will give results tailored to the interests of the searcher.

Unit 4.8 – Hardware Investigators

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Computer Science** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Different parts make up a computer.	• Name the different parts of a computer such as Hard Drive, RAM, Network Card etc. • Define what is meant by hardware, components and peripherals. • Describe the function of these different parts. • Show this knowledge by answering a quiz.

Unit 5.1 – Coding

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Computer Science** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Code can be simplified to complete the same process with less lines of code. Simplified code runs faster and uses less processing memory, it is said to be more efficient. Computer generated variables in 2Code are tags given to objects and these can be used to control object types meaning less lines of code are needed.	• Review a program that uses several objects of the same type and requires all of them to do the same thing such as having the speed set to 5. • Identify that common tags known as computer generated variables can used in the program to control all the objects. • Create a simplified code structure that functions exactly the same as the original code by using the common tags objects share. • Check both versions of the code work exactly the same way.

• A simulation is a model that represents a real or imaginary situation. Plans of an algorithm that represents a real or imaginary situation can be created and then used to program a simulation in 2Code.	• Explain what a simulation is and provide examples of physical systems that could be made into a simulation. • Plan an algorithm of a physical system such as traffic light sequences. • Consider in the algorithm plan essential details such as the sequence of traffic lights and when each light colour will display and for how long. • Convert the algorithm plan into a program within 2Code. • Test the program and how it compares to real-life. • Make adaptations to the program to consider variations in real-life situations. For example, adapting a simple traffic light sequence that needs adjusting because too much traffic is building up.
• The **timer every** command can be used to make code repeat forever.	• Recognise when the **timer every** command could be used in a suitable scenario to make code repeat forever. • Incorporate the **timer every** command into a program such as a traffic light sequence.
• 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.	• Use planning to create a program. • During planning, use decomposition to break down the plan into the key parts that are required to get the program functioning. • Consider if there are any unnecessary details in the plan that aren't essential for the functioning of the intended program (abstraction).

• Abstraction is a way of de-cluttering and removing unnecessary details to get a program functioning.	• Recognise what abstraction is and why it is important. • When planning a program, use abstraction to remove any unnecessary complications. • Realise in examples such as creating a traffic light sequence in 2Code, we can follow a process of abstraction to remove unnecessary details that aren't crucial to getting the program to function. For example, not including moving traffic, pavements or pedestrians.
• A function is a block or sequence of code that can be accessed when it is needed. This means code doesn't have to be rewritten every time it is needed. Instead, the function can be called each time it is needed.	• Recognise the **create function** command as part of the create and change variable group of blocks. • Insert a **create function** command into a program and name it. • Include code to a newly created function such as setting the position of an object. • Insert the **call function** command and assign the function created. • Include the **call function** command as part of an event, such as when the ball hits a wall, the function is called to reset its position back to the start.
• Strings are text or a combination of text characters and numbers within programs. An example could be a program that has a string type variable that is used to keep a player informed of their progress in a game.	• Recognise what a string is in a program and all instances of a string including how it is used. • Create a **string variable** and initialise it (give it a value). • Create code that changes the value of the string such as setting the string variable to a random word every 1 second. • Use the **print to screen** command to show how the string variable value is changing every 1 second.

• 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.

Unit 5.3 – Spreadsheets

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• A formula can be written in a sheet to convert units of length and distance.	• Explain what is meant by metric. • Write a simple formula for converting cm to m and m to cm using cell references, • Drag a formula from one cell to adjoining cells. • Complete similar task for metres to km.
• A spreadsheet can be used to model a real-life problem, in this case the area and perimeter of shapes.	• Explain what is meant by 'modelling'. • Write a simple formula to work out area. • Write a simple formula to work out perimeter. • Use the formulae to solve a problem.
• A spreadsheet can be used to investigate a problem such as the frequency of a number rolled on a collection of die.	• Add dice to the sheet. • Use the count tool to count the number rolled on the dice and apply this to a table. • Create a spreadsheet with formula. • Use the graphing functionality to display the results on screen.
• Spreadsheets can be created to support the organisation of real-life situations (school cake sale, ticket sales. pocket money spending, event organisation).	• Create spreadsheets to appropriately model the chosen scenarios. • Use formulae to analyse the data.

	• Use the spreadsheet to answer questions and make decisions.
• A spreadsheet tool can be used to investigate if a hypothesis is true.	• Define what is meant by a variable in a spreadsheet. • Use the count tool to count the number of vowels in a text. • Answer the hypothesis that the letter ‘e’ is the most popular letter in English. • Solve another problem using the count tool.

Unit 5.4 – Databases

NATIONAL CURRICULUM LINKS.	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• A database can be used to search for information.	• Open an existing database in 2Investigate. • Click on a record and see how the information is entered. • Enter data using words and numbers as well as drop down menus. • Use drop down menus to make the data entry more efficient. • Sort, group and arrange information in a database. • Search for information in a database. • Display information in tabular format and chart form. • Answer questions involving the interrogation of a database.
• Users can contribute to a collaborative database.	• Create an avatar for use in the database and pick out key information they could record in it. • Enter their data into the database. • Look at the collaborative and completed database. • Ask three questions to encourage their peers to interrogate the database.

• Databases can be created to cover a range of topics or themes. .	• Choose a suitable topic for a database. • Set up the database with appropriate fields. • Ass at least 8 records to the database. • Write five questions using their database for their peers to answer. • Use databases created by their peers to answer questions.

Unit 5.5 – Game Creator

NATIONAL CURRICULUM LINKS.	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• It is important to plan out a game before commencing on making it.	• Evaluate other games against criteria prior to beginning the creative process highlighting what works well and what could be improved. . • Use a design document to set the scene of the game. • Research what would make appropriate textures for aspects of the game and save these to the design document.
• A game design program has specific functions for the designer to use.	• Use the key functions of the game creator tool. • Remember the importance of saving regularly. • Design and add appropriate graphical elements to their game including floor, walls and ceiling. • Consider the appropriate places to locate game hazards which make the game more interesting and add to playability. • Add in game music to support the game theme. • Commence on making their game.

• The design of characters and quest items is a key aspect of game creation.	• Design the quest item and add in movement, sound effects and actions. • Consider where to place the quest items so it is possible to finish the game, and everything is collectible. • Place the enemies in the game in such a way as to provide challenge but not make it impossible to play. • Use their knowledge to create at least three levels.
• A finished game must be playable and possible for the player to complete.	• Finish making the game. • Write clear instructions that set a scene and provide gameplay instructions for the user. • Share the game online so other people can play it.
• Evaluation is important so a game can be improved and made more playable and exciting.	• Evaluate games made by their peers using given criteria. • Read evaluation of their game from other. • Make appropriate improvements to their game.

Unit 5.6 – 3D Modelling

NATIONAL CURRICULUM LINKS.	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• 3D modelling can be done via a computer program.	• Define what a 3D model is. • Select a design template in the 3D modelling program. • Select a viewpoint – Net, Points, 3D – depending upon what part of the design process they are carrying out. • Rotate a 3D model to see what the model looks like from a variety of angles. • Used the fill tool to add texture to their design. • Deigned a house of their dreams using the 3D program.
• Moving points changes the appearance of a 3D model.	• Select a vehicle design template from the program. • Change the location of the points to alter the appearance of the model. • Add textures to the model using the painting tools. • Use the design slider to alter the width of the model. • Design a vehicle to meet a design brief.

• A 3D design program can be used to meet a design brief.	• Add and remove points on a model. • Use a regular polygon and then add points to the model to change its appearance. • Design a model to meet a design brief – a piece of packaging for holding something.
• Models need refining before they are printed out using a standard printer or 3D printer.	• Refining a model is important prior to the final printing process. • Know what a STL file is. • Print their model onto paper/card or via a 3D printer. • Construct their 3D model if appropriate.

Unit 5.7 – Concept Maps

NATIONAL CURRICULUM LINKS.	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• There is a need for visual representation when generating and discussing complex ideas. This can be represented in the form of a concept map.	• Define a concept as an idea in the form of a question. • Begin to think about concept maps as a tool for organising and representing knowledge in a web. • Take part in an activity to make a physical concept map. • Develop a checklist of rules for creating a concept map.
• A computer program can be used to create a concept map.	• Set up a concept map using a blank template. • Add nodes to the map understanding they represent concepts or ideas. • Connect nodes together using a connection line and that arrows can be used to organise the date on the concept map. • Create a concept map linked to a specific topic. • Evaluate what worked well in the concept map and how it could be improved.
• A concept map can be used to retell information and stories.	• Open the story mode in the concept map program that allows the nodes to be used as a basis for their writing. • Create a piece of text using nodes as a basis for their writing.
• Collaborative concept maps allow many users to contribute to the same map and therefore quickly and easily share ideas.	• Make a concept map collaborative and save the document in the correct folder.

	• Contribute to a collaborative concept map on a given theme.

Unit 5.8 – Word Processing (WORD)

NATIONAL CURRICULUM LINKS.	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that:**	**Procedural** – By the end of the unit the students **will know how to:**
• A word processing tool can be used to create a range of documents.	• Open a blank document or a premade template. • Navigate around the toolbar at the top of the screen. • Expand the toolbar categories to give additional options. • Save a document into a specific folder. • Use the SHIFT or CAPS LOCK to write capital letters. • Select specific words to format. • Type sentences and then format them accordingly. • Format a document so it is easy to read.
• Images can be added to a document.	• Switch between portrait and landscape mode. • Use appropriate keyboard shortcuts for copy, paste and cut. • Search for images online. • Download images to a document. • Correctly reference the owner of the images.
• Images can be edited in Word using Word Wrap.	• Edit an image using the image handles. • Crop and resize an image. • Wrap text around an image. • Alter the transparency of an image.

• The look of text within a document can be changed.	• The text formatting toolbar allows the text within the document to be changed. • Apply a style to the document. • Add in headings and subheadings to a document. • Use a range of bullet points including numbered lists.
• Various features within the program will enhance the documents look and usability.	• Add in drop capitals. • Insert text boxes and shapes to a document. • Layer objects within a document. • Add in hyperlinks to a document to link to an external website.
• Tables can be used to present information within a document.	• Add in WordArt to a document in a range of styles. • Insert a table. • Merge and unmerge table cells. • Add in columns and rows to a table. • Distribute rows and columns in a table. • Choose table borders. • Change the background colour of a cell in a table.
• A template can be used to create a document.	• Look at the range of templates in the program. • Talk about the advantages of using a template over a blank document. • Create a document using a template. • Use the spelling and grammar check.
• Page layout can be improved by using headings and columns.	• Look at the layout features of a newspaper. • Insert columns into a blank document. • Use columns and learning from previous lessons to create a newspaper front page.

Unit 6.1 – Coding

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Computer Science** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Number elements combined with a number variable and an **if/else** statement can be used to create an onscreen countdown timer. Selection can be achieved through the use of **if/else** statements.	• Create a number element within design view of 2Code and name it. • Create a **number variable** command, name it and give it a value e.g. 30 to represent 30 seconds. • Use a **timer every** command set at 1 second with an **if/else** statement nested within it. • Create a sequence of code within the **if/else** statement that subtracts the number variable value by 1 each time. • Ensure **else** is selected if the condition reaches 0 seconds. • Use an **alert** command within **else** that gives an onscreen message. • Incorporate a **restart** command at the end of the **if/else** statement.

• The coordinates of objects can be used in code such as moving the position of them. The position of an object on the screen in 2Code is referenced using x and y coordinates.	• Identify the coordinates of any given object within 2Code by referencing the attributes table when an object is clicked on within design view. • Set an object's x and y coordinates. • Set an object's x and y coordinates to equal those of another object e.g. within a game when a bug is clicked on it is set to hide and a splat image appears at the same coordinates.
• The **launch command** can be used within 2Code to open another Purple Mash file or an external website when it is called in a program.	• Identify the **launch** command as part of the control group of commands within 2Code. • Incorporate the **launch** command within a program such as nested within an event. • Assign an activity or choose a webpage for the **launch** command.
• 2Code contains tabs in the coding view. Tabs can be used to help organise code.	• Recognise tabs in the code view of 2Code. • Label tabs in the code view of 2Code. • Organise key parts of code into their relevant tabs. • Recognise how tabs are useful when creating programs and during the debugging process.
• Using functions helps with making programs more efficient. Instead of writing the same sequence of code repeatedly when needed, a function can be created and called when required. Procedures are an independent piece of code. In 2Code, a procedure might be coded as a function.	• Recognise the **create function** command as part of the create and change variable group of blocks. • Recognise that a code within a function such as code that intends to draw a square is known as a procedure as it is independent of the main body of code. • Create a function within 2Code that makes an onscreen turtle draw regular shapes when called x number of times, extending beyond creating a square from Year 5.

• Flowcharts can represent procedures within a program. Flowcharts can be referenced when a program is executed to test whether a program is running as expected according to the flowchart.	• Interpret flowcharts which show procedures for a simulation. • From interpreting flowcharts which show procedures, predict how a program will run when specific events are met. • Run a program that represents the procedures shown in flowcharts. • Test a program alongside procedures on a flowchart to see whether the program is running as expected.
• Input is defined by information going into a computer. Inputs could consist of pressing a key, swiping a screen, clicking an object on the screen with a mouse or typing using the keyboard. All of these can be used to increase user interaction within a program. **Prompt for input** and **get input** are both defined as user input.	• Recognise the command blocks used for obtaining input in 2Code. • Explore the **prompt for input** and **get input** commands including knowing their difference and how they could be used within a program. • Use the **prompt for input** command within a program as part of joining strings together. Remember that when strings are combined it is referred to a concatenation. • Use the **get input** command as part of an **if/else** statement. For example, if input equals boo the sheep moves right off screen, else the sheep moves left.

• Text 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 pre-made text adventure. • Recognise why there is selection in the program and what happens when a condition is met or isn't met based on user input. • Interpret the purpose of variables within the text adventure and when their value is changed from 0 to 1. • See how functions are used for each room within the program and know when they are called. • Interpret how the **repeat until** command is used to run the program until a variable attributed to it is set to 1. • Plan own text adventures and adapt given code to support in creating own text adventure.

Unit 6.4 – Blogging

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• A blog is an online vehicle for displaying thoughts and ideas in an informal style.	• Explain the difference between a blog and vlog. • Talk about the key features of a blog home page. • Talk about what makes a good blog post.
• It is important to plan out the theme and content of a blog before writing it.	• Plan out a blog post on a given theme using a concept map.
• People can contribute to blogs by adding their own posts.	• Open the blogging tool. • Create their own blog. • Use the planning from the last lesson to write a blog post. • Publish the post.
• Blog posts written by others can be commented on.	• Wait for their blog post to be approved by the teacher. • Add a comment to a blog post written by another student.

Unit 6.5 – Text Adventures

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Computer Science** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• A text adventure is a computer game that uses text instead of graphics.	• Play a simple text-based adventure game on their device. • Plan out a text adventure game using a concept mapping tool. • Print out the concept map ready for the next lesson.
• Concept map plans for a story adventure can be used to plan the text-based adventure game.	• Use 2Create a Story to create an e-book text-based adventure. • Use the overview button to see how the pages link together. • Evaluate and then share their game with their peers.
• It is important to have a good level of coding comprehension in order the understand how a text adventure works.	• Follow the flow of a program on screen and follow it on a paper alternative. • Identify when commands are executed. • Identify variables in a program and what their function is in a text adventure program. • Identify all instances **if** commands. • Explain the purpose of a control loop including what selections it is checking for and which function is called.

• Debugging is a key part of coding and essential if code is to run properly.	• Debug a piece of code for a text-based adventure game. • Use the information from the debugging process to improve the code and add extra functionality.

Unit 6.6 – Networks

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Computer Science** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• The difference between the World Wide Web and the Internet.	• Carry out a survey with people over the age of 25 about the Internet and communication. • Recall the different ways they use the Internet at home and school. • Recall the difference between the Internet and World Wide Web. • Answer a quiz to test their knowledge.
• LAN and WAN are different kinds of networks,	• Talk about all the connected devices they use in school and at home. • Talk about wired and wireless networks and the key hardware needed for this. • Explain the difference between LAN and WAN. .
• The Internet has changed our lives in many ways.	• Give examples of well-known search engines. • Explain what an ip address is. • Write about the history of the Internet and Tim Berners-Lee • Consider how the Internet has changed things on their lifetime.

Unit 6.7 – Quizzing

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• The level, interests and capability of the audience need to be considered when making a game for younger children.	• Contribute to a collaborative concept map about what makes a good quiz and the different types of quizzes there are. • Make the introduction screen appealing with clear instructions for the user. • Add and edit images into the game. • Design a quiz to meet a specific brief and then share the game with a wider audience.
• A good quiz to appeal to younger students and their peers should have a range of different question types.	• Use the range of question types in the software to create a quiz on 10 questions with a diverse range of questions. • Add in a front screen with clear instructions and sounds. • Share the quiz for others to play.
• There are a range of software tools for creating quizzes to improve grammar skills.	• Play a range of grammar skill games. • Create a quiz for grammar skill consolidation. • Evaluate their quiz and quizzes produced by others.
• A quiz can be made to teach students how to interrogate a database.	• Recall what a database is. • Complete a ready-made quiz using a ready-made database. • Write a quiz of their own that involves searching a database.

• A range of questions can be used to produce a quiz linked to a curriculum area.	• Design a curriculum based game show style quiz which is used to challenge the teachers to see if they are smarter than a 10 or 11 year old. • Used a range of question types.

Unit 6.8 – Understanding Binary

NATIONAL CURRICULUM LINKS.	**Dominant strand for this unit: Computer Science** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• Binary is a number system using only 1 and 0 and is how data in a computer is saved and used.	• Recall that anything that puts information into a computer is called an input. • Recall that the number system we use is based around 10 integers and known as denary. • Explain that computers use a binary system based around 2 integers - 0 and 1 and 0 refers to off and 1 refers to on. • Complete a puzzle based around the binary system.
• All denary numbers can be represented in binary. For example counting in binary from zero to 15, or writing a friend's age in binary	• Complete a sentence explaining that for a computer to understand and interpret information everything must be converted to binary. • Begin representing numbers in binary format. • Convert simple binary to denary and vice versa. • Answer a quiz to test their knowledge. • Use a program to convert their age to binary.

• It is possible to represent the state of an object in a game as active or inactive using the respective binary values of 1 or 0.	• Complete a guided coding activity looking at switching objects on and off. • Make a simple program where objects are on or off. • Complete a quiz to test their knowledge of binary.

Unit 6.9 – Spreadsheets with Microsoft Excel

NATIONAL CURRICULUM LINKS	**Dominant strand for this unit: Information Technology** • 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. **There will be elements from the other two strands due to the nature of the computing curriculum.**

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **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, and 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.
• A spreadsheet can be used to model a situation.	• Define what is meant by a computational model. • Use the sum feature when typing a formula. • Format cells as currency. • Use a computational model in a real-life situation.

• Excel 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. • How to use flash fill to speed up data entry. • Define a delimiter and be aware of the different delimiter options. • Split cells for ease. • Sort data in a sheet using the appropriate feature.
• Formulae can be used for percentages, averages, max and min 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.
• A spreadsheet can be created to model a real-life situation.	• Understand the advantages of using formulae when data is liable to change. • Plan an event using a spreadsheet.
• Spreadsheets can be used to solve a given problem.	• Use the skills gathered over the unit to solve a series of problems. • Look at how to print a spreadsheet or a selection of the data.