Computing – Progression Document

Year	Computing Systems	Creating Media A	Programming A	Data and Information	Creating Media B	Programming B
EYFS	**Understanding technology:** Identify forms of IT in the classroom and school. Experiment with digital devices to see how they work showing curiosity. Ask questions about devices. Role-play with devices using real-world experiences. Express an opinion about what they like and don't like about digital devices. Play maths games on a device. Start, stop and pause music videos/song files. Enjoy dancing along to a music video/song file. Watch a video online to learn a new skill. Begin to identify why technology might be used.	**Expressing ideas:** Use art programmes/apps to express creativity and communicate ideas. Use programmes/apps to create own music. Use programmes/apps to record sounds for a purpose. Use programmes/apps to create and retell simple stories. Create a simple slide show and talk it through as a presentation. Record photos/short videos for a purpose.	**Coding activities:** Follow simple instructions. Give simple instructions for others to follow. Problem solve in the real world. Use simple coding apps to accomplish tasks. Build resilience and persistence when using coding apps. Problem solve when using coding apps.	**Data:** Use programmes/apps to communicate with others. Explore simple maths programmes such as creating graphs/collecting data. Add own data to a simple graph app, such as favourite colour. Use an app to store data such as a shopping list when role-playing.	**See Creating Media A**	**See Programming A**
1	**Technology around us** Explain how some technology examples help us. Explain technology as something that helps us. Locate examples of technology in the classroom. Name the main parts of a computer. Switch on and log into a computer.	**Digital painting** Draw lines on a screen and explain which tools have been used. Make marks on a screen and explain tools used. Use the paint tools to draw a picture. Make marks with the square and line tools. Use the shape and line tools effectively.	**Moving a robot** Match a command to an outcome. Predict the outcome of a command on a device. Run a command on a device. Follow an instruction. Give directions. Recall words that can be acted out. Compare forwards and backwards movements.	**Grouping data** Describe objects using labels. Identify the label for a group of objects. Match objects to groups. Count a group of objects. Count objects. Group objects. Describe an object. Describe a property of an object. Find objects with similar properties.	**Digital writing** Identify and find keys on a keyboard. Open a word processor. Recognise keys on a keyboard. Enter text into a computer. Use backspace to remove text. Use letter, number, and space keys. Explain what a variety of keys do.	**Programming animations** Compare different programming tools. Find which commands to move a sprite. Use commands to move a sprite. Run own program. Use a Start block in a program. Use more than one block by joining them together. Change a value.

	Use a mouse to click and drag. Click and drag to make objects on a screen. Use a mouse to create a picture. Use a mouse to open a program. Save work to a file. Say what a keyboard is for. Type their name on a computer. Delete letters. Open work from a file. Use the arrow keys to move the cursor. Discuss how we benefit from rules Give examples of some of these rules. Identify rules to keep us safe and healthy when we are using technology in and beyond the home.	Use the shape and line tools to recreate the work of an artist. Choose appropriate shapes. Create a picture in the style of an artist. Make appropriate colour choices. Choose appropriate paint tools and colours to recreate the work of an artist. Say which tools are helpful and why. Know that different paint tools do different jobs. Change the colour and brush sizes. Make dots of colour on the page. Use dots of colour to create a picture in the style of an artist. Explain that pictures can be made in lots of different ways. Say whether they prefer painting using a computer or using paper. Spot differences between painting on a computer and on paper.	Predict the outcome of a sequence involving forwards and backwards commands. Start a sequence from the same place. Compare left and right turns. Experiment with turn and move commands to move a robot. Predict the outcome of a sequence involving up to four commands. Choose the order of commands in a sequence. Debug a program. Explain what a program should do. Identify several possible solutions. Plan two programs. Use two different programs to get to the same place.	Count how many objects share a property. Group objects in more than one way. Group similar objects. Choose how to group objects. Describe groups of objects. Record how many objects are in a group. Compare groups of objects. Decide how to group objects to answer a question. Record and share what they have found.	Identify the toolbar and use bold, italic, and underline. Type capital letters. Change the font. Select all of the text by clicking and dragging. Select a word by double-clicking. Decide if changes have improved writing. Say what tool has been used to change the text. Use 'undo' to remove changes. Explain the differences between typing and writing. Make changes to text on a computer. Say why they prefer typing or writing.	Find blocks that have numbers. Say what happens when values are changed. Add blocks to sprites. Delete a sprite. Show that a project can include more than one sprite. Choose appropriate artwork for a project. Create an algorithm for each sprite. Decide how each sprite will move. Add programming blocks based on own algorithm. Test programs they have created. Use sprites that match their design.
2	**IT around us** Describe some uses of computers. Identify examples of computers. Identify that a computer is a part of IT. Identify examples of IT.	**Digital photography** Explain what they did to capture a digital photo. Recognise what devices can be used to take photographs. Talk about how to take a photograph.	**Robot algorithms** Choose a series of words that can be enacted as a sequence. Follow instructions given by someone else. Give clear instructions. Show the difference in outcomes between two	**Pictograms** Compare totals in a tally chart. Record data in a tally chart. Represent a tally count as a total. Enter data onto a computer.	**Digital music** Describe music using adjectives. Identify simple differences in pieces of music. Say what they do and don't like about a piece of music. Create a rhythm pattern.	**Programming quizzes** Identify that a program needs to be started. Identify the start of a sequence. Show how to run own program. Change the outcome of a sequence of commands.

	Identify that some IT can be used in more than one way. Sort IT by what it's used for. Find examples of information technology. Sort IT by where it is found. Talk about uses of information technology. Demonstrate how IT devices work together. Recognise common types of technology. Say why we use IT. List different uses of information technology. Say how rules can help keep us safe. Talk about different rules for using IT. Explain the need to use IT in different ways. Identify the choices made when using IT. Use IT for different types of activities.	Explain the process of taking a good photograph. Explain why a photo looks better in portrait or landscape format. Take photos in both landscape and portrait format. Discuss how to take a good photograph. Identify what is wrong with a photograph. Improve a photograph by retaking it. Experiment with different light sources. Explain why a picture may be unclear. Explore the effect that light has on a photo. Explain choices. Recognise that images can be changed. Use a tool to achieve a desired effect. Apply a range of photography skills to capture a photo. Identify which photos are real and which have been changed. Recognise which photos have been changed.	sequences that consist of the same commands. Use an algorithm to program a sequence on a floor robot. Use the same instructions to create different algorithms. Compare own prediction to the program outcome. Follow a sequence. Predict the outcome of a sequence. Explain the choices they made for own mat design. Identify different routes around own mat. Test own mat to make sure that it is usable. Create an algorithm to meet own goal. Explain what own algorithm should achieve. Use own algorithm to create a program. Plan algorithms for different parts of a task. Put together the different parts of own program. Test and debug each part of the program.	Use a computer to view data in a different format. Use pictograms to answer simple questions about objects. Explain what the pictogram shows. Organise data in a tally chart. Use a tally chart to create a pictogram. Answer 'more than'/'less than' and 'most/least' questions about an attribute. Create a pictogram to arrange objects by an attribute. Tally objects using a common attribute. Choose a suitable attribute to compare people. Collect the data they need. Create a pictogram and draw conclusions from it. Give simple examples of why information should not be shared. Share what they have found out using a computer. Use a computer program to present information in different ways.	Explain that music is created and played by humans. Play an instrument following a rhythm pattern. Connect images with sounds. Relate an idea to a piece of music. Use a computer to experiment with pitch. Explain how own music can be played in different ways. Identify that music is a sequence of notes. Refine own musical pattern on a computer. Add a sequence of notes to own rhythm. Create a rhythm which represents an animal they've chosen. Create own animal's rhythm on a computer. Explain how they changed own work. Listen to music and describe how it makes them feel. Review own work.	Match two sequences with the same outcome. Predict the outcome of a sequence of commands. Build the sequences of blocks they need. Decide which blocks to use to meet the design. Work out the actions of a sprite in an algorithm. Choose backgrounds for the design. Choose characters for the design. Create a program based on the new design. Build sequences of blocks to match own design. Choose the images for own design. Create an algorithm. Compare own project to own design. Debug own program Improve own project by adding features.
3	**Connecting computers** Explain that digital devices accept inputs. Explain that digital devices produce outputs. Follow a process. Classify input and output devices.	**Stop-frame animation** Create an effective flip book-style animation. Draw a sequence of pictures. Explain how an animation/flip book works.	**Sequencing sounds** Explain that objects in Scratch have attributes (linked to). Identify the objects in a Scratch project (sprites, backdrops).	**Branching databases** Create two groups of objects separated by one attribute. Investigate questions with yes/no answers.	**Desktop publishing** Explain the difference between text and images. Identify the advantages and disadvantages of using text and images.	**Events and actions in programs** Choose which keys to use for actions and explain own choices. Explain the relationship between an event and an action.

	Describe a simple process. Design a digital device. Explain how to use digital devices for different activities. Recognise similarities between using digital devices and non-digital tools. Suggest differences between using digital devices and non-digital tools. Discuss why we need a network switch. Explain how messages are passed through multiple connections. Recognise different connections. Demonstrate how information can be passed between devices. Explain the role of a switch, server, and wireless access point in a network. Recognise that a computer network is made up of several devices. Identify how devices in a network are connected. Identify networked devices around them. Identify the benefits of computer networks.	Create an effective stop-frame animation. Explain why little changes are needed for each frame. Predict what an animation will look like. Break down a story into settings, characters and events. Create a storyboard. Describe an animation that is achievable on screen. Evaluate the quality of own animation. Review a sequence of frames to check own work. Use onion skinning to help make small changes between frames. Evaluate another learner's animation. Explain ways to make own animation better. Improve own animation based on feedback. Add other media to own animation. Evaluate own final films. Explain why they added other media to own animation.	Recognise that commands in Scratch are represented as blocks. Choose a word which describes an on-screen action for own plan. Create a program following a design. Identify that each sprite is controlled by the commands they choose. Create a sequence of connected commands. Explain that the objects in own project will respond exactly to the code. Start a program in different ways. Combine sound commands. Explain what a sequence is. Order notes into a sequence. Build a sequence of commands. Decide the actions for each sprite in a program. Make design choices for own artwork. Identify and name the objects they will need for a project. Implement own algorithm as code. Relate a task description to a design.	Make up a yes/no question about a collection of objects. Arrange objects into a tree structure. Create a group of objects within an existing group. Select an attribute to separate objects into groups. Group objects using own yes/no questions. Select objects to arrange in a branching database. Test own branching database to see if it works. Compare two branching database structures. Create yes/no questions using given attributes. Explain that questions need to be ordered carefully to split objects into similarly sized groups. Create a physical version of a branching database. Create questions that will enable objects to be uniquely identified. Independently create questions to use in a branching database. Create a branching database that reflects own plan. Suggest real-world uses for branching databases. Work with a partner to test own identification tool.	Recognise that text and images can communicate messages clearly. Change font style, size, and colours for a given purpose. Edit text. Explain that text can be changed to communicate more clearly. Create a template for a particular purpose. Define the term 'page orientation'. Recognise placeholders and say why they are important. Choose the best locations for own content. Make changes to content after I've added it. Paste text and images to create a magazine cover. Choose a suitable layout for a given purpose. Identify different layouts. Match a layout to a purpose. Compare work made on desktop publishing to work created by hand. Identify the uses of desktop publishing in the real world. Say why desktop publishing might be helpful.	Identify a way to improve a program. Choose a character for own project. Choose a suitable size for a character in a maze. Program movement. Choose blocks to set up own program. Consider the real world when making design choices. Use a programming extension. Build more sequences of commands to make own design work. Choose suitable keys to turn on additional features. Identify additional features (from a given set of blocks). Match a piece of code to an outcome. Modify a program using a design. Test a program against a given design. Evaluate own project. Implement own design. Make design choices and justify them.

	The Internet	Audio production	Repetition in shapes	Data logging	Photo editing	Repetition in games
4	Demonstrate how information is shared across the internet. Describe the internet as a network of networks. Discuss why a network needs protecting. Describe networked devices and how they connect. Explain that the internet is used to provide many services. Recognise that the World Wide Web contains websites and web pages. Describe how to access websites on the WWW. Describe where websites are stored when uploaded to the WWW. Explain the types of media that can be shared on the WWW. Explain that internet services can be used to create content online. Explain what media can be found on websites. Recognise that they can add content to the WWW. Explain that there are rules to protect content. Explain that websites and their content are created by people. Suggest who owns the content on websites. Explain that not everything on the World Wide Web is true.	Explain that the person who records the sound can say who is allowed to use it. Identify the input and output devices used to record and play sound. Use a computer to record audio. Discuss what sounds can be added to a podcast. Inspect the soundwave view to know where to trim own recording. Re-record own voice to improve own recording. Explain how sounds can be combined to make a podcast more engaging. Plan appropriate content for a podcast. Save own project so the different parts remain editable. Improve own voice recordings. Record content following own plan. Review the quality of own recordings. Arrange multiple sounds to create the effect they want. Explain the difference between saving a project and exporting an audio file. Open own project to continue working on it. Choose appropriate edits to improve own podcast.	Create a code snippet for a given purpose. Explain the effect of changing a value of a command. Program a computer by typing commands. Test own algorithm in a text-based language. Use a template to create a design for own program. Write an algorithm to produce a given outcome. Identify everyday tasks that include repetition as part of a sequence, eg brushing teeth, dance moves. Identify patterns in a sequence. Use a count-controlled loop to produce a given outcome. Choose which values to change in a loop. Identify the effect of changing the number of times a task is repeated. Predict the outcome of a program containing a count-controlled loop. Explain that a computer can repeatedly call a procedure. Identify 'chunks' of actions in the real world. Use a procedure in a program. Design a program that includes count-controlled loops.	Choose a data set to answer a given question. Identify data that can be gathered over time. Suggest questions that can be answered using a given data set. Explain what data can be collected using sensors. Identify that data from sensors can be recorded. Use data from a sensor to answer a given question. Identify the intervals used to collect data. Recognise that a data logger collects data at given points. Talk about the data that they have captured. Explain that there are different ways to view data. Sort data to find information. View data at different levels of detail. Plan how to collect data using a data logger. Propose a question that can be answered using logged data. Use a data logger to collect data. Draw conclusions from the data that they have collected. Explain the benefits of using a data logger.	Explain why they might crop an image. Improve an image by rotating it. Use photo editing software to crop an image. Experiment with different colour effects. Explain that different colour effects make you think and feel different things. Explain why they chose certain colour effects. Add to the composition of an image by cloning. Identify how a photo edit can be improved. Remove parts of an image using cloning. Experiment with tools to select and copy part of an image. Explain why photos might be edited. Use a range of tools to copy between images. Choose suitable images for own project. Create a project that is a combination of other images. Describe the image they want to create. Combine text and own image to complete the project. Review images against given criteria. Use feedback to guide making changes.	List an everyday task as a set of instructions including repetition. Modify a snippet of code to create a given outcome. Predict the outcome of a snippet of code. Choose when to use a count-controlled and an infinite loop. Modify loops to produce a given outcome. Recognise that some programming languages enable more than one process to be run at once. Choose which action will be repeated for each object. Evaluate the effectiveness of the repeated sequences used in own program. Explain what the outcome of the repeated action should be. Explain the effect of own changes. Identify which parts of a loop can be changed. Re-use existing code snippets on new sprites. Develop own design explaining what own project will do. Evaluate the use of repetition in a project. Select key parts of a given project to use in own design. Build a program that follows own design.

	Explain why they need to think carefully before they share or reshare content. Explain why some information online may not be honest, accurate, or legal.	Listen to an audio recording to identify its strengths. Suggest improvements to an audio recording.	Develop own program by debugging it. Make use of own design to write a program.	Interpret data that has been collected using a data logger.		Evaluate the steps they followed when building own project. Refine the algorithm in own design.
5	**Systems and searching** Describe that a computer system features inputs, processes, and outputs. Explain that computer systems communicate with other devices. Explain that systems are built using a number of parts. Explain the benefits of a given computer system. Identify tasks that are managed by computer systems. Identify the human elements of a computer system. Compare results from different search engines. Make use of a web search to find specific information. Refine a web search. Explain why we need tools to find things online. Recognise the role of web crawlers in creating an index. Relate a search term to the search engine's index. Explain that a search engine follows rules to rank results	**Video production** Compare features in different videos. Explain that video is a visual media format. Identify features of videos. Experiment with different camera angles. Identify and find features on a digital video recording device. Make use of a microphone. Capture video using a range of filming techniques. Review how effective own video is. Suggest filming techniques for a given purpose. Create and save video content. Decide which filming techniques they will use. Outline the scenes of own video. explain how to improve a video by reshooting and editing. Select the correct tools to make edits to own video. Store, retrieve, and export own recording to a computer. Evaluate own video and share own opinions.	**3D Modelling** Add 3D shapes to a project. Move 3D shapes relative to one another. View 3D shapes from different perspectives. Lift/lower 3D objects. Recolour a 3D object. Resize an object in three dimensions. Duplicate 3D objects. Group 3D objects. Rotate objects in three dimensions. Accurately size 3D objects. Combine a number of 3D objects. Show that placeholders can create holes in 3D objects. Analyse a 3D model. Choose objects to use in a 3D model. Combine objects in a design. Construct a 3D model based on a design. Explain how own 3D model could be improved. Modify own 3D model to improve it.	**Flat-file databases** Create a database using cards. Explain how information can be recorded. Order, sort, and group own data cards. Choose which field to sort data by to answer a given question. Explain what a field and a record is in a database. Navigate a flat-file database to compare different views of information. Combine grouping and sorting to answer specific questions. Explain that data can be grouped using chosen values. Group information using a database. Choose multiple criteria to answer a given question. Choose which field and value are required to answer a given question. Outline how 'AND' and 'OR' can be used to refine data selection. Explain the benefits of using a computer to create charts.	**Introduction to vector graphics** Discuss how vector drawings are different from paper-based drawings. Experiment with the shape and line tools. Recognise that vector drawings are made using shapes. Explain that each element added to a vector drawing is an object. Identify the shapes used to make a vector drawing. Move, resize, and rotate objects they have duplicated. Explain how alignment grids and resize handles can be used to improve consistency. Modify objects to create a new image. Use the zoom tool to help me add detail to own drawings. Change the order of layers in a vector drawing. Identify that each added object creates a new layer in the drawing. Use layering to create an image.	**Selection in quizzes** Identify conditions in a program. Modify a condition in a program. Recall how conditions are used in selection. Create a program with different outcomes using selection. Identify the condition and outcomes in an 'if... then… else...' statement. Use selection in an infinite loop to check a condition. Design the flow of a program which contains 'if… then… else…' Explain that program flow can branch according to a condition. Show that a condition can direct program flow in one of two ways. Identify the outcome of user input in an algorithm. Outline a given task. Use a design format to outline own project. Implement own algorithm to create the first section of own program. Share own program with others. Test own program.

	Give examples of criteria used by search engines to rank results. Order a list by rank. Describe some of the ways that search results can be influenced. Explain how search engines make money. Recognise some of the limitations of search engines.	Make edits to own video and improve the final outcome. Recognise that own choices when making a video will impact on the quality of the final outcome.		Refine a chart by selecting a particular filter. Select an appropriate chart to visually compare data. Ask questions that will need more than one field to answer. Present own findings to a group. Refine a search in a real-world context.	Copy part of a drawing by duplicating several objects. Recognise when they need to group and ungroup objects. Reuse a group of objects to further develop own vector drawing. Compare vector drawings to freehand paint drawings. Create a vector drawing for a specific purpose. Reflect on the skills they have used and why they have used them.	Extend own program further. Identify the setup code they need in own program. Identify ways the program could be improved.
6	**Communication and collaboration** Describe how computers use addresses to access websites. Explain that internet devices have addresses. Recognise that data is transferred using agreed methods. Explain that all data transferred over the internet is in packets. Explain that data is transferred over networks in packets. Identify and explain the main parts of a data packet. Explain that the internet allows different media to be shared. Recognise how to access shared files stored online. Send information over the internet in different ways.	**Web page creation** Discuss the different types of media used on websites. Explore a website. Know that websites are written in HTML. Draw a web page layout that suits own purpose. Recognise the common features of a web page. Suggest media to include on own page. Describe what is meant by the term 'fair use'. Find copyright-free images. Say why they should use copyright-free images. Add content to own web page. Evaluate what own web page looks like on different devices and suggest/make edits. Preview what own web page looks like.	**Variables in games** Explain that the way a variable changes can be defined. Identify examples of information that is variable. Identify that variables can hold numbers or letters. Explain that a variable has a name and a value. Identify a program variable as a placeholder in memory for a single value. Recognise that the value of a variable can be changed. Decide where in a program to change a variable. Make use of an event in a program to set a variable. Recognise that the value of a variable can be used by a program. Choose the artwork for own project.	**Spreadsheets** Collect data. Enter data into a spreadsheet. Suggest how to structure own data. Apply an appropriate format to a cell. Choose an appropriate format for a cell. Explain what an item of data is. Construct a formula in a spreadsheet. Explain which data types can be used in calculations. Identify that changing inputs changes outputs. Apply a formula to multiple cells by duplicating it. Calculate data using different operations. Create a formula which includes a range of cells.	**Selection in physical computing** Create a simple circuit and connect it to a microcontroller. Explain what an infinite loop does. Program a microcontroller to make an LED switch on. Connect more than one output component to a microcontroller. Design sequences that use count-controlled loops. Use a count-controlled loop to control outputs. Design a conditional loop. Explain that a condition is either true or false. Program a microcontroller to respond to an input. Explain that a condition being met can start an action. Identify a condition and an action in own project.	**Sensing movement** Apply own knowledge of programming to a new environment. Test own program on an emulator. Transfer own program to a controllable device. Determine the flow of a program using selection. Identify examples of conditions in the real world. Use a variable in an if, then, else statement to select the flow of a program. Experiment with different physical inputs. Explain that checking a variable doesn't change its value. Use a condition to change a variable.

	Explain how the internet enables effective collaboration. Identify different ways of working together online. Recognise that working together on the internet can be public or private. Choose methods of communication to suit particular purposes. Explain the different ways in which people communicate. Identify that there are a variety of ways to communicate over the internet. Compare different methods of communicating on the internet. Decide when they should and should not share information online. Explain that communication on the internet may not be private.	Describe why navigation paths are useful. Explain what a navigation path is. Make multiple web pages and link them using hyperlinks. Create hyperlinks to link to other people's work. Evaluate the user experience of a website. Explain the implication of linking to content owned by others.	Create algorithms for own project. Explain own design choices. Choose a name that identifies the role of a variable. Create the artwork for own project. Test the code that they have written. Identify ways that own game could be improved. Share own game with others. Use variables to extend own game.	Apply a formula to calculate the data they need to answer questions. Explain why data should be organised. Use a spreadsheet to answer questions. Produce a chart. Suggest when to use a table or chart. Use a chart to show the answer to questions.	Use selection (an 'if…then…' statement) to direct the flow of a program. Create a detailed drawing of own project. Describe what own project will do. Identify a real-world example of a condition starting an action. Test and debug own project. Use selection to produce an intended outcome. Write an algorithm that describes what own model will do.	Explain the importance of the order of conditions in else, if statements. Modify a program to achieve a different outcome. Use an operand (e.g. <>=) in an if, then statement. Decide what variables to include in a project. Design the algorithm for own project. Design the program flow for own project. Create a program based on own design. Test own program against own design Use a range of approaches to find and fix bugs.