Year 2 Computing

Key Vocabulary

Digital Literacy		
National Curriculum Principles	**Objectives**	**Knowledge and key Vocabulary**
Recognise common uses of information technology beyond school.	- To understand that information comes from a variety of sources; e.g. books, websites, TV… - To understand that technology allows you to quickly access information from a range of sources. - To understand how digital technology is used at home and school. - To be able to explore a range of electronic information as part of a topic.	- With increasing independence, allow children to navigate websites or information stored on a website, seesaw or other platform (paper or digital). Such as CBBC Newsround, National Geographic or sites linked to current projects. - With increasing independence, recognise that icons, menus, hyperlinks are used to organise information (Seesaw) - Have access to different types of information from a range of sources (books, websites, TV) - Recognise digital technologies in everyday life. To link to current projects e.g Great Fire of London. How is technology used in the Fire services? Difference in technology from then to now.
Use technology safely and respectfully, keeping personal information private.	- To understand that anyone, from anywhere, can access the internet. - To understand that personal information should not be shared online.	- With increasing independence, understand that the internet is a network of servers that sends and receives information. - https://www.thinkuknow.co.uk - http://www.safetynetkids.org.uk/personal-safety/staying-safe-online/ - https://www.bbc.co.uk/cbeebies/grownups/article-internet-use-and-safety
identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.	To understand what to do when worried about something and to recognise specific places to get help - CEOP.	Share the school child friendly online safety policy with children. Talk about trusted adults at home and who to share their concerns with.

Year 2 Computing

Information Technology		
What's the big picture? To continue to upskill our children to have the foundations of skills within IT and digital literacy to ensuring they are fully prepared for future work, opportunities within technology and the ability to be safe online.		
National Curriculum Principles	**Objectives**	**Knowledge and key Vocabulary**
To use, with support, technology purposefully to create , organise, store, manipulate and retrieve digital content. (Ongoing)	● Organise folders and documents on their iPads and to understand the importance. ● To understand that files can be uploaded organised to help with retrieval of digital content. ● To understand that digital content can come in a variety of forms: images, videos and audio.	● Show children how to add apps to a folder for a purpose. ● Upload documents to Seesaw and add to a folder. ● Use Clips/GarageBand/Camera to create to show a variety of digital content.

Year 2 Computing

Computer Science		
National Curriculum Principles: (Ongoing) ● 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. These National Curriculum objectives are delivered through the use of Everyone Can Code **(Get Started with Code 1)**. The lessons have been designed to develop and build on skills. Children in year 2 will use Tynker app.		
Lesson	**Objectives**	**Knowledge and key Vocabulary**
o	● Recap the concept of coding. ● Understand the goals of the sessions/ outcome. ● Understand the purpose of a working wall. ● With increasing independence, learn how to use Seesaw.	To recap to the terms coding, developer and apps (applications). Chn to understand that Seesaw will be used to share their work. (P13) Complete optional app design activity. Discuss terms researchers, designers, coders and communicators. Chn to brainstorm apps that could help solve a problem (problems in the classroom or at home).
1	**Sequence** - Using everyday examples, describe what sequences are. - Construct a sequence based on a familiar story. - Code using sequences.	Introduction: As a class, recap what a sequence is. Identify and order steps in an everyday routine eg brushing your teeth using prompts. First, next, last. What would happen if the steps were out of order? Activity: Using The Gingerbread Man or similar story chn to draw the story and then arrange themselves in order so the story is sequenced correctly. Class to reorder to ensure the story is correct.

Year 2 Computing

			Practise: Complete practise activity on Tynker (P18) Optional activity: Play 'App I Spy' (P20) Explain that apps are programs that run on devices or computers to make something happen or help us do something. Discuss how all apps are designed with a purpose in mind, and that different apps do different things. After a few practice rounds of App I Spy, start organizing the apps into folders according to their purpose. While playing, the class can come up with new folder categories. Journal: Using audio recording and Seesaw app, chn explain what is a sequencing, why it is important to follow the steps and why we use sequencing.
2	**Sequence and command** - Build a step-by-step sequence. - Understand the importance of order when sequencing instructions. - Code using sequence.		Introduction: Continue to develop understanding of the term sequence by creating a dance sequence and then teach other students how to perform it. Recapping the understanding of a command. Activity: Challenge students to design their own crazy dance sequence using these dance moves from the Keynote (P23). Send the Keynote file to them using AirDrop. They should reorder the slides to create their unique dance. Explain that when we give commands to computers, they always perform them in the order given. Practise: Complete the practise activity on Tynker. Optional app design activity: Explain to the class that apps can be designed differently to achieve different learning goals. Compare the different approaches to supporting learning, using a Venn diagram two different apps that the children are familiar with (P26) Journal: Using an audio recording, chn explain:

Year 2 Computing

Key Vocabulary

		- How did their partner know how to do the dance? - What would happen if they mixed up the sequence? - Why is sequencing important when you're telling a computer what to do?
3	**Sequence and command** - Understand that some steps within a sequence can be reordered and still achieve the same outcome. - Construct a flexible sequence and compare it with a partner's work. - Identify which parts of the sequence are step-by-step and which parts are flexible. - Code using different sequences to achieve the same outcome.	Introduction: Explore flexible sequences, which are sequences that can be completed in more than one way. Activity: Discuss the Keynotes slides of a sequencing event. Ask students to work in pairs with the Keynote slides to show a different sequence for getting dressed. Ask the pairs to share the sequences. To tell a computer what to do, we need to identify and sequence steps. Sometimes these steps can be put in any order. Coders call this flexible sequencing. Practise: Complete practise activity using Tynker(P31) Optional app design activity: Support students to understand how we can design apps to help us engage with other people in our community. Explore two apps with the class, one that broadcasts information—such as Twitter—and another that shares information with a specific audience—BBC News. Explain to students how important it is to be aware of the audience for any information they share online (P33). Journal: Audio recording that explains the following: - What did they learn about flexible sequencing? - How is flexible sequencing different from step-by-step sequencing? - When do they use flexible sequencing in their lives?"
4	**Loops** - Understand what a loop is. - Identify where a loop can make an instruction more efficient. - Understand why a loop is	Introduction: Tell students, "When we're coding, if I want to repeat an action, that's called looping." Ask the class, "Can you think of any other everyday activities where you repeat or loop an instruction? Activity: Introduce the following body percussion sounds one at a time using the Keynote slides provided.

Year 2 Computing

	powerful. - Code with loops.		Demonstrate the sound and have students repeat it. Introduce students to the loop symbol and show how you add the loop symbol slide before the moves that need to be repeated. Select the move to be performed, deleting the other images, and add the number of repetitions to the loop slide, then skip the slides that are the same moves. As a class, practice performing the new song. Then have students develop their own songs using the Keynote file. They should use the camera on the iPad to record their performance. Compare the act of copying and pasting all the slides each time to just adding the loop symbol. Which one is easier? Practise: Practise using Tynker, Children to complete puzzles in Space Cadet lesson 4. Ensure children are using repeating patterns, such as walking a certain number of times or walking, then jump. When the children figure out repeating sequence, they can shorten their program by putting that code into a loop. Journal: Chn to upload a recording/ screenshot of their keynote song sequence, adding an explanation of what a loop is and when they'd use one. Optional app design activity: Chn to reflect on their experience using Tynker. Thinking about what they like/dislike. Chn to edit a screenshot annotating the image. (p39)
5	**Debugging.** - Describe what debugging is. - Demonstrate the use of debugging in an everyday situation. - Debug code.		Introduction: Ask chn to tell you how they complete an everyday activity such as tying your shoelaces. Follow their instructions exactly, even if things go wrong. Allow the children time to self-correct and modify their instructions until they are able to accomplish the goal. (P41) Activity: Using the resources from page 42, chn are to use a floor grid to help solve problems. One child to be a robot and the class to be the computer. Chn to use a range of code to enable the robot to navigate through the maze. (More detail on p42)

Year 2 Computing

		Practise: Complete puzzles in Space Cadet Lesson 5. For this lesson the code is already completed, the children have to debug to find errors and then fix. Journal: Children are to upload their floor grids and code cards to Seesaw. They are to add an audio recording to explain their experience. - How did they control the robot? - How did their group debug the code? - Why is debugging important?
6	**Events and actions.** - Understand that an event us an action that causes something to happen. - Recognise that events give us options in coding - they cause things to happen only when the event occurs. - Express an event in words and symbols. - Code using events and action.	Introduction: Play statues with the children. Get them to move around and when you turn around the chn are to stop - if they move then they are out. This is to help the children understand that the teacher turning around is an 'event' and the children freezing is the 'action'. Activity: Children are to create a robot remote control. (See supporting material page 48.) Practise: Children to complete puzzles in Space Cadet Lesson 6. This lesson children will explore 4 events: touch, collision, tilt and messaging. Ensure the program is running for these events to work. Journal: Chn to upload images of the events and a short video of their robots to Seesaw.
7	**IF statements.** - Understand that we can make actions occur only under certain conditions. - Use IF statements in	Introduction: Discuss with the class the rules of our classroom. Then repeat the rules as IF statements. For example: If we want to talk, we raise our hand. If someone is having trouble, we help them. Activity:

		everyday life and in coding. - Code using IF statements.	Chn to play Simon Says based on IF statements. As a class play Simon Says. Remind the class they are to only follow the instruction IF Simon Says first. Chn to then work in groups to describe a familiar game as a series of IF statements and to create a video to explain this to other students. Practise: Complete Space Cadet Lesson 7. They will use IF statements to run code if a certain condition is true. IF statements all students to make their program react to changing or uncertain conditions. Journal: Children are to add their explain-a-game to their Seesaw folder and to explain what an IF statement is.
8	**Algorithms.** - Write an algorithm to solve a problem. - Design a simple program.		Introduction: An algorithm is a series of steps to solve a problem. To ensure an algorithm works we need to ensure we are precise with our instructions. As a class follow a simple recipe. Explain to the children that in coding we call the step-by-step instructions an algorithm. Activity: Solve the maze. Children are to create their own maze and then create an algorithm to allow their character to exit the maze without hitting any hazards. Practise: Complete all puzzles in Space Cadet lesson 8. They will combine all skills they have learnt. Journal: Children are to upload their maze images to Seesaw noting their experience. - What did they enjoy about creating their maze? - What was easy/ difficult?

Year 2 Computing
Key Vocabulary

Glossary	
Algorithm	An algorithm is a set of steph-by-step rules or instructions.
Bug	A bug is an error in your code.
Coding	Coding is telling a computer what to do.
Command	A command is a specific action.
Conditional statement or action	Conditional statements or actions occur only under certain conditions.
Debugging	Debugging is the process of identifying and fixing errors.
Developer	Developers write code to build their own apps and games.
Event	An event is an action that causes something else to happen.
Loop	A loop is an instruction to repeat a set of commands for a specific number of times.
Sequence	A sequence is the order in which things happen, like patterns and events.