**Year 6 Computing Curriculum – Spring Term 2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Theme: Sensing | | | | | | | | |
| **Curriculum objectives** | | | **Vocabulary** | | | | | **Links across the curriculum** |
| - 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  - 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 | | | **Keyword** | Definition | sequences | a pattern or process in which one thing follows another. | | [**National curriculum links**](https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study)   * 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 * 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 | |
| design | to think up and plan out in the mind | decompose | Break down into smaller and manageable ‘chunks’ | |
| Count-controlled loop | A count-controlled loop is a form of repetition in which a set of commands are carried out a specific number of times. | Condition-controlled loop | A condition-controlled loop is a form of repetition in which a set of commands stop being carried out when a condition is met. | |
| debug | to fix | conditions | Conditions are statements that need to be met for a set of actions to be carried out | |
| commands | to order or instruct | program | a plan of what will be done | |
| Infinite loop | An infinite loop is a loop that commands the instruction/set of instructions to repeat forever. | algorithms | a determined and finite procedure for solving a problem | |
| **Prior Knowledge:**  EYFS – To follow two step instructions. Year 1 – Commands for a robot. Year 2 – plan and debug algorithm  Year 3 - Sequencing Sounds Year 4 – repetition in Shapes  Year 5 - Selection In physical computing | | | | |  | | | |
| **Lesson Sequence** | | **Key Knowledge** | | | | | **Key Skills** | |
| 1 The micro:bit | | Pupils will be introduced to the micro:bit as an input, process, output device that can be programmed. Pupils will familiarise themselves with the device itself and the programming environment, before creating their own programs. They will then run their programs on the device.  **Note:** This unit is written assuming that you will be using a desktop or laptop computer (not a tablet) to connect micro:bits. | | | | | To create a program to run on a controllable device   * I can apply my knowledge of programming to a new environment * I can test my program on an emulator * I can transfer my program to a controllable device | |
| 2 Go with the flow | | * Pupils will explore how if, then, else statements are used to direct the flow of a program. They will initially relate if, then, else statements to real-world situations, before creating programs in MakeCode. They will apply their knowledge of if, then, else statements to create a program that features selection influenced by a random number to create a micro:bit fortune teller project. | | | | | To explain that selection can control the flow of a program   * I can identify examples of conditions in the real world * I can use a variable in an if, then, else statement to select the flow of a program * I can determine the flow of a program using selection | |
| 3 Sensing inputs | | * Pupils will initially use the buttons to change the value of a variable using selection. They will then develop their programs to update the variable by moving their micro:bit using the accelerometer to sense motion. Finally, they will learn that a variable’s value remains the same after it has been checked by the program. | | | | | To update a variable with a user input   * I can use a condition to change a variable * I can experiment with different physical inputs * I can explain that checking a variable doesn’t change its value | |
| 4 Finding your way | | * Pupils will apply their understanding of the importance of order in programs. They will then use operands in selection to determine the flow of a program. Pupils will then modify a program which will enable the micro:bit to be used as a navigational device. To code this, they will adapt the code they completed to make a basic compass. | | | | | To use an conditional statement to compare a variable to a value   * I can use an operand (e.g. <>=) in an if, then statement * I can explain the importance of the order of conditions in else, if statements * I can modify a program to achieve a different outcome | |
| 5 Designing a step counter | | * Pupils will be working at the design level. They will pick out features of a step counter, a piece of technology with which they are likely to be familiar. They will then relate those features to the sensors on a micro:bit. In the main activity, pupils will design the algorithm and program flow for their step counter project. | | | | | To design a project that uses inputs and outputs on a controllable device   * I can decide what variables to include in a project * I can design the algorithm for my project * I can design the program flow for my project | |
| 6 Making a step counter | | * Pupils will use the design that they have created in Lesson 5 to make a micro:bit-based step counter. First they will review their plans, followed by creating their code. Pupils will test and debug their code, using the emulator and then the physical device. To successfully complete this project, Pupils will need to demonstrate their understanding of all the programming lessons they've had so far. | | | | | To develop a program to use inputs and outputs on a controllable device   * I can create a program based on my design * I can test my program against my design * I can use a range of approaches to find and fix bugs | |
| **Themes and links** | | | | | | | | |
| **Computing themes** | **Where these are covered:** | | | | | | | |
| **Technology around us**  Autumn 1 | * Scratch links to the real world and computer games the children know. | | | | | | | |
| **Digital painting**  Autumn 2 | * Design, make and evaluate process | | | | | | | |
| **Programming A**  Spring 1 | * the concept of selection in programming using the Scratch programme | | | | | | | |
| **Data /information**  Spring 2 | * Storing the commands and the effect on language on the outcome of your commands. | | | | | | | |
| **Creating media**  Summer 1 | * Your own designs of Scratch | | | | | | | |
| **Programming B**  Summer 2 | * Using scratch to implement an algorithm as a code | | | | | | | |