# Computing – Year 5

| Spring 2 | Selection in quizzes | | | | | |
|---|---|---|---|---|---|---|
| **Prior learning** | This unit assumes that learners will have prior experience of programming using block-based construction (e.g. Scratch), understand the concepts of 'sequence' (Year 3 units: Sequencing Sounds and Events and actions in programs) and 'repetition' (Year 4 units: Repetition in shapes and Repetition in games), and have some experience of using 'selection'. Ideally, learners will have completed 'Programming A – Selection in physical computing' before undertaking this unit, as this will provide them with the required knowledge of 'selection'. | | | | | |
| **Lesson objective** | To explain how selection is used in computer programs | To relate that a conditional statement connects a condition to an outcome | To explain how selection directs the flow of a program | To design a program that uses selection | This can be merged into one lesson is necessary<br>To create a program that uses selection | To evaluate my program |
| **Key vocabulary** | Selection, condition, true, false, count-controlled loop | Selection, condition, true, false, outcomes, conditional statement (the linking together of a condition and outcomes), algorithm, program, debug | Selection, condition, true, false, outcomes, question, answer, algorithm, program, debug | Task, design, algorithm, input, program, selection, condition, outcomes | Implement, design, algorithm, program, selection, condition, outcome, test, run | Design, algorithm, program, debug, test, setup, selection, condition, outcome |
| **Creative context** | Connect to careers in coding, game design, app development, and software testing.<br>Cross-curricular links: Maths (logical reasoning and sequences), English (writing clear questions and instructions), Science (using inputs and outputs to explore experiments). | | | | | |
| **Substantive knowledge** | I know that selection allows a program to make decisions based on conditions.<br>I know that conditions control the flow of a program.<br>I know that modifying a condition changes the action carried out. | I know that conditions can have two outcomes: true or false.<br>I know that selection can be repeated using loops to continually check a condition.<br>I know how conditional statements connect a condition to its outcome(s). | I know that program flow can branch depending on user input.<br>I know that a binary question can be represented as a condition in a program.<br>I know that algorithms can be used to plan program flow. | I know that a program's outcomes depend on the conditions set for user input.<br>I know that design templates help plan algorithms before coding.<br>I know that clear requirements guide program design. | I know that programs must be tested to ensure conditions produce the correct outcomes.<br>I know that debugging is necessary when a program does not behave as intended.<br>I know that sharing programs allows feedback to improve design. | I know that programs can be improved based on user input and experience.<br>I know that setup blocks ensure consistent experiences for all users.<br>I know that programs can be extended with additional features for usability. |

# Computing – Year 5

| | | | | | | |
|---|---|---|---|---|---|---|
| **Disciplinary knowledge** | I can identify conditions in a program.<br>I can modify a condition in Scratch.<br>I can predict the outcome of a program based on its conditions. | I can use an infinite loop to repeatedly check a condition.<br>I can create a program using if… then… else… to produce different outcomes.<br>I can identify conditions and their associated outcomes in a program. | I can design a branching algorithm with if… then… else… structure.<br>I can implement user input as conditions in Scratch.<br>I can test that both outcomes are achievable in my program. | I can outline a task and plan the program using a design format.<br>I can identify expected outcomes of user input in my design.<br>I can organise conditions and actions logically in a quiz algorithm. | I can implement a section of my algorithm in Scratch.<br>I can test my program and debug when needed.<br>I can share my program with peers and use feedback to improve it | I can identify improvements to make in my program.<br>I can implement setup code to standardize outcomes for all users.<br>I can extend my program to include further functionality. |
| **Recorded learning** | Pupils will explore pre-built Scratch programs to see how conditions control actions and try modifying conditions to observe changes in outcomes. | Pupils will create a program that respond differently to true and false conditions. | Pupils will design and implement algorithms that branch based on user input and test their programs to ensure both true and false outcomes work correctly. | Pupils will plan their interactive quiz, identifying questions, conditions, and outcomes for correct and incorrect answers. | Pupils will create the first section of their quiz, run and test it, and debug errors to ensure the outcomes match their design. | Pupils will evaluate their quizzes, make improvements to conditions and setup, extend their programs, and reflect on what worked well and what could be improved. |
| **Outcome for unit** | Unit outcome: Pupils will create a fully functional interactive quiz in Scratch that uses selection to control outcomes based on user input. This should be saved in their scratch accounts.<br><br>End of unit evaluation – Please complete the teacher/self evaluation slide for this unit (the slide is in the folder), by writing their names in the correct boxes. | | | | | |
| **Future learning** | In year 6, there are other data and information units which knowledge taught in this topic will help with. Further to this, it will also help with within other subjects like science or maths when looking into information gathering and statistics. | | | | | |