|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Term Pentecost 2**  **Subject Computing Year 5 Medium Term Planning** | | | | | | |
| **National Curriculum Objectives**  ● 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 | | | | | | |
|  | Lesson 1 | Lesson 2 | Lesson 3 | Lesson 4 | Lesson 5 | Lesson 6 |
| **Learning intention for each lesson:** | 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 | To create a program that uses selection | To evaluate my program |
| **Recall and retrieval** | *.*Revisit ‘select’ and identify how ‘conditions’ are used to control the flow of actions in a program.  Selection, condition, true, false, count-controlled loop. | Revisit the need to use repetition in selection to ensure that conditions are repeatedly checked. (Y4) | What will the outcome be from a simple program. | Follow a branching program- Identify the error. | Identify the outcome of user input in an algorithm | What is good about this program and one improvement. |
| **Sequence of knowledge throughout the lesson**  **:**  **Key skills within each lesson** | **Key knowledge**  Recall how conditions are used in selection  Identify conditions in a program  Modify a condition in a program  **Begin to know blocks in programs using the Scratch programming.**  **Modify the conditions in an existing program** | **Key knowledge**  Use selection in an infinite loop to check a condition  Identify the condition and outcomes in an ‘if… then… else…’ statement  Ceate a program that uses selection to produce different outcomes  **use the ‘if… then… else...’ structure in algorithms and programs.**  **Identify the two outcomes in given programs.**  **Explain how the condition informs which outcome will be selected.**  **Write their own programs that use selection with two outcomes.** | **Key knowledge**  Explain that program flow can branch according to a condition  Design the flow of a program that contains ‘if… then… else…’  Show that a condition can direct program flow in one of two ways  **Use the ‘if… then... else…’ structure can be used to identify two responses to a binary question (one with a ‘yes or no’ answer).**  **Identify that the answer to the question is the ‘condition’.**  **Use algorithms with a branching structure to represent the actions that will be carried out.**  **Learn how questions can be asked in Scratch, and how the answer control the outcomes.**  **Use an algorithm to design a program that uses selection to direct the flow of the program based on the answer provided.**  **Implement algorithm as a program and test.** | **Key knowledge**  Outline a given task  Use a design format to outline my project  Identify the outcome of user input in an algorithm  **Use selection to control the outcomes in an interactive quiz. Outline the requirements of the task.**  **Use an algorithm to show how they will use selection in the quiz to control the outcomes based on the answer given.**  **lComplete designs using design templates to identify the questions that will be asked, and the outcomes for both correct and incorrect answers.** | **Key knowledge**  Implement my algorithm to create the first section of my program  Test my program  Share my program with others  **Use the Scratch program to implement the first section of their algorithm as a program.**  **Run the first section of program to test.**  **Continue implementing their algorithm as a program.**  **Use and give feedback on others quizzes.** | **Key knowledge**  Identify ways the program could be improved  Identify the setup code I need in my program  Extend my program further  **Evaluate and identify ways to avoid problems.**  **Consider how the outcomes may change the program for subsequent users,**  **Identify and make improvements to own quiz as a result of playing other quizzes.** |
| **Scaffolding** | Support to use blocks | Support for writing a program. | Simple branching program. | Fewer questions | Support to evaluate | Simple improvements. |
| **Challenge** | identify the impact this changes have on the program. | More than 2 program. | More complex branching program. | More questions | Positives and a way to improve in feedback. | Use step up. |
|  | **Key Vocabulary**  Selection, condition, true, false, count-controlled loop | **Key Vocabulary**  Selection, condition, true, false, outcomes, conditional statement (the linking together of a condition and outcomes), algorithm, program, debug | **Key Vocabulary**  Selection, condition, true, false, outcomes, question, answer, algorithm, program, debug | **Key Vocabulary**  Task, design, algorithm, input, program, selection, condition, outcomes | **Key Vocabulary**  Implement, design, algorithm, program, selection, condition, outcome, test, run | **Key Vocabulary**  Design, algorithm, program, debug, test, setup, selection, condition, outcome |