ALMOND HILL JUNIOR SCHOOL MEDIUM TERM PLAN

TOPIC TITLE/SUBJECT: Computing - Programming B – Selection in quizzes **YEAR GROUP: 5** **TERM: Summer**

Vocabulary	Skills	What we already know
Algorithm Program Code Selection Condition Repetition Loop Branch Flow chart Binary questions Controlled outcomes Debug Setup	-To explain how selection is used in computer programs (recall how conditions are used in selection/identify conditions in a program/modify conditions in a program) -To relate that a conditional statement connects a condition to an outcome (use selection in an infinite loop to check a condition/identify conditions and outcomes in an 'if... then... else...' statement/ create a program that uses selection to produce different outcomes) -To explain how selection directs the flow of a program (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 2 ways) - To design a program that uses selection (outline a given task/use a design format to outline my project/identify the outcome of user input in an algorithm) -To create a program that uses selection (implement an algorithm to create a first section/test programs/share with others) -To evaluate my program (identify improvements/identify the setup code needed/extend programs further).	This unit assumes that learners will have prior experience of programming using block-based construction (e.g. Scratch, Logo), understand the concepts of 'sequence' and 'repetition', and have some experience of using 'selection'.

Application/Outcomes

Exploring conditions: Revisit previous learning on 'selection' and identify how 'conditions' are used to control the flow of actions in a program. Introduced to the blocks for using conditions in programs using the Scratch programming environment. Modify the conditions in an existing program & identify the impact this has.

*Selecting outcomes: D*evelop their understanding of selection by using the 'if… then… else...' structure in algorithms and programs. Revisit the need to use repetition in selection to ensure that conditions are repeatedly checked. Identify the two outcomes in given programs and how the condition informs which outcome will be selected. Learners use this knowledge to write their own programs that use selection with two outcomes.

*Asking questions: C*onsider how 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' and use algorithms with a branching structure to represent the actions that will be carried out if the condition is true or false. Learn how questions can be asked in Scratch, and how the answer, supplied by the user, is used in the condition to 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 their algorithm as a program and test whether both outcomes can be achieved.

Designing a quiz: Learners will be provided with a task: to use selection to control the outcomes in an interactive quiz. They will outline the requirements of the task and use an algorithm to show how they will use selection in the quiz to control the outcomes based on the answer given. Learners will complete their designs by using design templates to identify the questions that will be asked, and the outcomes for both correct and incorrect answers. To demonstrate their understanding of how they are using selection to control the flow of the program, they will identify which outcomes will be selected based on given responses.

*Testing a quiz: U*se the Scratch programming environment to implement the first section of their algorithm as a program. Run the first section of their program to test whether they have correctly used selection to control the outcomes and debug their program if required. They will then continue implementing their algorithm as a program. Once completed, they will consider the value of sharing their program with others so that they can receive feedback. Conclude the lesson by using another learner's quiz and providing feedback on it.

*Evaluating a quiz: I*dentify ways in which the program can be improved. Focus on issues where answers like those in the condition are given as inputs and identify ways to avoid such problems. Consider how the outcomes may change the program for subsequent users and identify how they can make use of 'setup' to provide all users with the same experience. Implement their identified improvements. Conclude the unit by identifying how they met the requirements of the given task and identifying the aspects of the program that worked well, those they improved, and areas that could improve further.

Other/Cross Curricular Links	Adaptation for SEND		
Potentially any quiz that requires specific subject knowledge – binary questions only	Adapted tasks	Adapted resources	Additional support