

Year 7 Computer Science Learning Outcomes

Unit 1: Programming Essentials in Scratch Part 1

- Compare how humans and computers understand instructions (understand and carry out)
- Define a sequence as instructions performed in order, with each executed in turn
- Predict the outcome of a simple sequence
- Modify a sequence
- Define a variable as a name that refers to data being stored by the computer
- Recognise that computers follow the control flow of input/process/output
- Predict the outcome of a simple sequence that includes variables
- Trace the values of variables within a sequence
- Make a sequence that includes a variable
- Define a condition as an expression that will be evaluated as either true or false
- Identify that selection uses conditions to control the flow of a sequence
- Identify where selection statements can be used in a program
- Modify a program to include selection
- Create conditions that use comparison operators (>,<,<=)
- Create conditions that use logic operators (and/or/not)
- Identify where selection statements can be used in a program that include comparison and logical operators
- Define iteration as a group of instructions that are repeatedly executed
- Describe the need for iteration
- Identify where count-controlled iteration can be used in a program
- Implement count-controlled iteration in a program
- Detect and correct errors in a program (debugging)
- Independently design and apply programming constructs to solve a problem (subroutine, selection, count-controlled iteration, operators, and variables)

Unit 2: Programming Essentials in Scratch Part 2

- Define a subroutine as a group of instructions that will run when called by the main program or other subroutines
- Define decomposition as breaking a problem down into smaller, more manageable subproblems
- Identify how subroutines can be used for decomposition
- Identify where condition-controlled iteration can be used in a program
- Implement condition-controlled iteration in a program
- Evaluate which type of iteration is required in a program
- Define a list as a collection of related elements that are referred to by a single name
- Describe the need for lists

- Identify when lists can be used in a program
- Use a list
- Decompose a larger problem into smaller subproblems
- Apply appropriate constructs to solve a problem