|  |
| --- |
| **Programming basics**  |
| Algorithm  | sequence of steps taken to complete a task |
| Storing data  | Input data as a variable or a constant and store it. |
| Processing  | Programs manipulate data with logical processes |
| Printing Data  | Data can be output using the print statement |

|  |
| --- |
| **Data Types**  |
| Integer  | An integer is a whole number (not a fraction) it is positive, negative, or zero. |
| Real/Float  | Is written with a decimal point dividing the integer and fractional parts. |
| Boolean  | has two possible values: true or false (yes or no) |
| Character  | any letter, number or symbol on a computer. 5 String a sequence of character |

|  |
| --- |
| **Programming Constructs**  |
| Sequence  | A set of logical steps carried out in order. |
| Selection  | Where a decision must be made in programming using IF statements. |
| Count  | controlled Iteration repeatedly executes a section of code a fixed number of times |
| Condition  | controlled iteration repeatedly executes a section of code until a condition is met - or no longer met |

|  |
| --- |
| **Program Coding**  |
| Comment  | annotation in the code of a computer program |
| Indentation  | denotes code within the loop that is repeated |
| Nesting  | A programming construct is included within another. |
| Maths operators | are: +, -, /, \*. (plus, minus, division, multiplication) |

|  |
| --- |
| **Key Vocabulary**  |
| Algorithm  | is a sequence of steps that can be followed to complete a task |
| Sequence  | A set of logical steps carried out in order. |
| Selection  | Where a decision must be made in programming using IF statements. |
| Iteration  | repeatedly executes a section of code |
| Variable  | Is a storage location paired with a name, which contains a value? |
| Constant  | a value that cannot be altered by the program during normal execution |
| Data types  | an attribute of data which tells the computer how the programmer intends to use the data |
| Comment  | annotation in the code of a computer program |

|  |  |  |
| --- | --- | --- |
| **Handling Data Types:**  | Convert to integer  | int(x) |
| Convert to float  | float(x) | Convert to string  | str(x) |
| Convert to Boolean  | bool(x) | User input in python is always a string by default. |

Summary

**Programming** is writing computer code to create a program, in order to solve a problem. Programs consist of a series of instructions to tell a computer exactly what to do and how to do it. An **algorithm** is a set of instructions that describes how to get something done. It is crucial that the steps in an algorithm are sequenced and performed in the right order - otherwise the algorithm will not work correctly. Algorithms can be designed using **pseudocode** and **flow charts**. They are written using **statements** and **expressions**. There are three basic building blocks (constructs) to use when designing algorithms: **sequencing, selection and iteration**. We create programs to **implement** algorithms. Algorithms consist of steps, where programs consist of statements. In programming, iteration is often referred to as **‘looping’**, because when a program iterates it ‘loops’ to an earlier step. It is implemented using **FOR** and **WHILE** statements. Selection is implemented in programming using **IF** statements.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Sequencing** | **Selection** | **Iteration** |
| Computer programs use variables to store information. Variables could be used to store the score in a game, the number of cars in a car park or the cost of items on a till. They work in a similar way to algebra, where a letter in your code can stand for a number. | Sequencing is the specific order in which instructions are performed in an algorithm. Algorithms consist of instructions that are carried out (performed) one after another | Selection is a decision or question. At some point, a program may need to ask a question because it has reached a step where one or more options are available. Depending on the answer given, the program will follow a certain step and ignore the others. | Iteration is the process of repeating steps. Iteration allows us to simplify our algorithm by stating that we will repeat certain steps until told otherwise. This makes designing algorithms quicker and simpler because they don’t have to include lots of unnecessary steps |
|  |  |  |  |

