Knowledge Organiser 2.1: Algorithms

1. Computational Thinking

Abstraction	The process of removing unnecessary details and including only the relevant details.It is a method of computational thinking that focusses on what is important in problem solving
Decomposition	The process of breaking a complex problem down into smaller more manageable parts. Dealing with many different stages of a problem at once is much more difficult than breaking a problem down into a number of smaller problems and solving each, one at time.
Advantages of Program Decomposition	• Makes problems easier to solve. Different people can work on different parts of a problem at the same time… • reducing development time

2. Input Processes and Output

Inputs	• Anything which needs to be supplied to the program so it can meet its goals. • Often input by the user. • Consider an appropriate variable name and data type for the input.
Processes	• Consider what calculations need to be performed while the program is running. • Does data need to change formats or data types

3. Structure Diagrams

- Structure diagrams illustrate problem decomposition.
- They can be used for developers to understand a problem to code and to share with users during systems analysis.
- They are produced using a method known as step-wise refinement.
- Break problem down using decomposition into ever smaller components.

3. Flowcharts, Pseudocode and OCR Reference Language

Flowchart	A method of representing the sequences of steps in an algorithm in the form of a diagram. Sometimes called a Flow diagram
Structure Diagram	A diagram showing a top-down breakdown of a complex problem
Pseudocode	A text based alternative of representing the sequences of steps in an algorithm. Pseudo-code can be thought of as a simplified form of programming code.
OCR Reference Language	You must be able to read this but you can always use Python in your exams—but be precise

4. Types of Errors

Syntax Error	Syntax errors are errors which break the grammatical rules of the programming language. They stop it from being run/translated
Logic Errors	Logic errors are errors which produce unexpected output. On their own they

5. Trace Tables

- A vital skill for understanding program flow and testing the accuracy of an algorithm for logic is called “Tracing Execution”.
- Examine a printed extract of program code and running thorough the program.
- Take each line at a time and write out in a trace table the current state of each variable. Noting down any output the program produces.
- Each variable present in the program should have its own column in the trace table.
- A new row should be added under any column if the state of a variable changes.
- Trace tables are an excellent way to track down logic errors in a problem.