Definitions and Concepts for CAIE Computer Science IGCSE

Topic 7: Algorithm design and problem-solving

Program Development Life Cycle: The main stages involved in creating a program: analysis, design, coding, and testing.

Analysis: The stage where the problem is understood and requirements are identified, using abstraction, decomposition, and identification of inputs, processes, and outputs.

Abstraction: Removing unnecessary details to focus only on what is important.

Decomposition: Breaking a problem into smaller, more manageable sub-problems.

Design: Planning the solution before writing code, often using structure diagrams, flowcharts, or pseudocode.

Coding: Translating the design into a working program by writing program code and performing iterative testing.

Testing: Checking that the program works correctly and meets requirements, using test data.

System Decomposition: Breaking a system into sub-systems to make it easier to design, understand, and test.

Inputs: Data that goes into a system.

Processes: Actions or calculations performed on data.

Outputs: Information produced by the system.

Storage: Data saved for later use.

Structure Diagram: A diagram used to represent decomposition, showing tasks broken down into smaller sub-tasks.

Pseudocode: A text-based, language-independent way of describing a solution in structured steps, using keywords.

Flowchart: A diagram showing the step-by-step flow of a process using standard symbols.

Linear Search: A searching algorithm that checks items in a list one by one until the desired item is found or the list ends.

Bubble Sort: A sorting algorithm that repeatedly compares and swaps adjacent elements in a series of passes, causing the largest elements to “bubble” to the end with each pass.

Totalling: Adding up a running total of values using a loop.

Counting: Counting how many times a condition is met by increasing a counter each time.

Finding Minimum/Maximum: Identifying the smallest or largest value by comparing each new value to the current minimum or maximum.

Finding Average: Calculating the mean by totalling all values and dividing by the count.

Validation: Checking that input data is reasonable and sensible before acceptance.

Range Check: Ensures data is within a specified range.

Length Check: Ensures data has the correct number of characters.

Type Check: Ensures data is the correct data type.

Presence Check: Ensures data is not left blank.

Format Check: Ensures data follows a specific pattern.

Check Digit: An extra digit calculated from others, used to detect errors.

Verification: Checking that data matches the original source to prevent data entry errors.

Visual Check: User compares entered data with the original by eye.

Double Entry Check: Data is entered twice and compared.

Test Data: Data used to check that a program works correctly.

Normal Data: Typical, valid data within the expected range.

Abnormal Data: Invalid data that should be rejected.

Extreme Data: Largest and smallest acceptable values.

Boundary Data: Values at the limits (extreme) and just outside the limits.

Dry-Run: Manually going through an algorithm step by step without a computer.

Trace Table: A table used during a dry-run to record variable values, outputs, and prompts at each step, helping a user understand or test the flow of a program.

Syntax Error: An error caused by breaking the rules of the language, such as missing keywords or misspelling commands.

Logic Error: An error caused by incorrect logic, such as using the wrong operator or calculation, that will cause an incorrect result despite the program running.

Runtime Error: An error that occurs while the program is running, such as dividing by zero or using an invalid index.

Algorithm: A set of instructions that can be followed to perform a specific task.

Program Code: The actual code written in a specific programming language.