

# Definitions and Concepts for OCR Computer Science A-level

## Component 2.2: Problem Solving and Programming

---

### 2.2.1 Programming Techniques

**Branching:** A programming control structure where the code selects one or more alternative paths depending on the evaluation of a boolean expression.

**Functions:** A subroutine that can be called to perform a task or calculation and always return a value. Functions can be called in an expression, or be assigned to a variable.

**Global Variables:** A variable declared in the main program which exists outside any of the subroutines, but can be used anywhere in the program.

**Integrated Development Environment (IDE):** A software package that allows a user to develop and implement code more easily, with features for editing, debugging, version control, testing and compilation.

**Iterations:** A programming control structure where a set of statements is repeated either a fixed number of times or until a condition is met.

**Local Variables:** A variable declared within a subroutine of a program, which only exists and can be used within that subroutine.

**Modularity:** The technique of breaking down a complex problem into simpler, self-contained components called modules, where each module is an implementation of a specific subtask required to solve a problem.

**Object Oriented Programming:** A programming paradigm where the system is viewed as a set of objects, each with their own data (attributes) and procedures (methods), that can interact with each other. All processing is done by objects.

**Parameter passing by reference:** Passing the address or pointer of the required value into a procedure.

**Parameter passing by value:** Creating a temporary local copy of the actual value of a variable and passing it into the procedure.

**Parameters:** The data structures required to be passed into a subroutine.

**Procedures:** A subroutine that is called by simply writing its name in the code. Procedures do not have to return a value in the program.

This work by [PMT Education](https://www.pmt.education) is licensed under [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)



**Recursion:** A programming subroutine where a section of code calls itself until a base case is met. The base case must be chosen to avoid any possibility of an infinite loop.

**Sequences:** A programming control structure in which statements are executed one after another as they appear in the script.

**Subroutines:** A uniquely named section of code that is written to perform a specific task within a program.

### **2.2.2 Computational Methods**

**Backtracking:** An algorithm that incrementally finds a solution by methodically trying different sequences and abandoning a path when it knows it cannot lead to a valid solution.

**Computable Problems:** A problem for which every instance can be solved in a finite number of steps by means of algorithm.

**Computational Methods:** A method of solving a problem which by some form of computation in devising and implementing an algorithm.

**Data Mining:** An algorithm that finds a solution by analysing a large data sets to uncover trends and relationships between variables.

**Divide and Conquer:** An algorithm design technique to decompose and solve problems by reducing the problem size with each iteration, until the sub problem becomes solvable.

**Heuristics:** A 'rule of thumb' algorithm which can produce a valid albeit sub-optimal solution for a hard (intractable) problem as an approximation.

**Performance Modelling:** The process of simulating the behaviour of a model under different virtual user and system loads by mathematical approximation.

**Pipelining:** The process of splitting a task into parts and then searching for subtasks that can be processed simultaneously to overlap the processing of each part.

**Problem Decomposition:** The process of splitting a given problem into smaller, solvable sub-problems that are easier to conceive, understand, maintain and program.

**Problem Recognition:** The ability to recognise the most effective strategy to solve a problem.

**Visualisation:** The use of a visual representation of an algorithm or data structure to translate a problem and its solution to a more human readable form.

