

Definitions and Concepts for AQA Computer Science A-level

Topic 1: Fundamentals of Programming

1.1 Programming

1.1.1 Data Types

Arrays: A data structure for storing a finite, ordered set of data of the same data type within a single identifier.

Boolean: A data type that can only store one of two possible values (1 or 0, TRUE or FALSE etc.)

Character: A data type for storing a letter, number or special character..

Data Type: An attribute of data that determines what sort of data is being stored that tells the compiler how it will be used in a program.

Date/Time: A data type for storing date or time values in an identifiable format.

Integer: A data type for storing whole number values (positive or negative) with no decimal parts.

Language-Defined (Built-In) Data Types: A primitive data type provided by a programming language.

Pointer/Reference: A data type used as a store for memory addresses of objects created at runtime. †

Real/Float: A data type for storing numbers with decimal or fractional parts.

Records: A data structure that stores related data items in elements called fields, organised based on attributes.

String: A data type for storing a sequence of alphanumeric characters or symbols, typically within quotation marks.

User-Defined Data Types: Custom data types designed by the user by combining existing data types, for the bespoke needs of their 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/)



1.1.2 Programming Concepts

Assignment: A statement for giving a created variable a value that is consistent with its data type.

Constant Declaration: A statement for creating a constant in a program.

Iteration: A programming structure where a set of statements are repeated a fixed number of times (definite iteration), or until a condition is met (indefinite iteration).

Nested Iteration: A programming structure of placing iteration loops within other iteration loops.

Nested Selection: A programming structure of placing selection statements within other selection statements.

Selection: A programming structure for deciding which statements to perform next based on a certain condition or set of conditions.

Subroutines (Procedures/Functions): A uniquely named section of code that is written to perform a specific task within a program that can be called by using its name in a programming statement.

Variable Declaration: A statement for creating a variable in a program. The variable is defined with a name and sometimes a data type.

1.1.3 Arithmetic Operations in a Programming Language

Addition: An arithmetic operator that returns the sum of two numeric values (integer or floats).

Arithmetic Operator: An operator that takes two numeric values and performs some form of mathematical manipulation (adding, subtracting, multiplying, dividing) to return a numeric value.

Exponentiation: An arithmetic operator that raises a numeric value to the power of another numeric value and returns the result.

Integer Division: An arithmetic operator that divides a numeric value (integer or float) by another but returns an integer quotient.

Modulo: An arithmetic operator that divides a numeric value (integer or float) by another and returns the remainder.

Multiplication: An arithmetic operator that returns the product of two numeric values (integers or floats).



Real/Float Division: An arithmetic operator that divides a numeric value (integer or float) by another and returns a Real/Float.

Rounding: An arithmetic operator that converts a real/float to a more approximate representation that reduces the number of digits to represent the number by rounding up or down to a given number of decimal places.

Subtraction: An arithmetic operator that returns the difference between two numeric values (integers or floats).

Truncation: An arithmetic operator that reduces the number of digits to represent a real/float by cutting off a number after a given number of decimal places.

1.1.4 Relational Operations in a Programming Language

Equal To: A relational operator that returns TRUE if and only if the two values in question are the same.

Greater Than: A relational operator that returns TRUE if and only if the first value is larger than the second.

Greater Than Or Equal To: A relational operator that returns TRUE if the first value is larger than the second, or if they are equal.

Less Than: A relational operator that returns TRUE if and only if the first value is smaller than the second.

Less Than Or Equal To: A relational operator that returns TRUE if the first value is smaller than the second, or if they are equal.

Not Equal To: A relational operator that returns TRUE if and only if the two values in question are not the same.

Relational Operators: An operator that compares two values and returns TRUE or FALSE.

1.1.5 Boolean Operations in a Programming Language

AND (\wedge): A boolean operator which returns TRUE (or 1) if and only if all inputs are TRUE (or 1).

Boolean Operators: An operator that compares two boolean values and returns TRUE or FALSE (1 or 0).

NOT (\neg): A boolean operator which returns TRUE (or 1) if and only if the input is FALSE (or 0).

OR (\vee): A logical operator which returns TRUE (or 1) if and only if any one of the inputs are



TRUE (or 1).

XOR: A logical operator which returns TRUE (or 1) if and only if exactly 1 of the inputs are TRUE (or 1).

1.1.6 Constants and Variables in a Programming Language

Constants: A data item that, once declared, retains the same value for the entire duration of the program run.

Variables: A data item that, once declared, can be used as a short term memory container for a temporary value that may change over the duration of the program run.

1.1.7 String-Handling Operations in a Programming Language

Character Code: A binary representation of a character using the rules of a character set (eg. ASCII or Unicode). Characters can be converted to and from character code.

Concatenation: A function used to merge two strings into a single longer string.

Length: A function that returns the number of characters (including spaces) present in a given string.

Position: A function that returns the index used to locate and identify a given character in a string.

String Conversion Operations: Functions used to convert strings to and from other data types when possible; such as integers, floats and datetime.

Substring: A function used to slice a string to obtain the relevant substring embedded within the longer string.

1.1.8 Random Number Generation in a Programming Language

Random Number Generation: The process of generating a random number to be used by a program generally by inputting a value known as the seed into an algorithm.

1.1.9 Exception Handling

Exception Handling: The process of dealing with exceptional events triggered by circumstances that are inconsistent with what the program was coded for. This is done by informing the user of the error that has occurred and how it can be rectified in the next run.

1.1.10 Subroutines (Procedures/Functions)

Functions: A subroutine that can be called to perform a task or operation and always returns a value. They can be called in an expression or be assigned to a variable.



Procedures: A subroutine that can be called by simply writing its name in the code. Procedures do not have to return values in a program.

Subroutines (Procedures/Functions): A uniquely named section of code that is written to perform a specific task within a program that can be called by using its name in a programming statement.

1.1.11 Parameters of Subroutines

Interface: Code used to describe the data type and characteristics of data being passed when a subroutine calls another subroutine.

Parameters: The identified data being passed into describing a subroutine. Parameters contain the arguments (data items) that will be used by the subroutine when it is called.

1.1.13 Local Variables in Subroutines

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

1.1.14 Global Variables in Subroutines

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

1.1.15 Role of Stack Frames in Subroutine Calls

Stack Frames: A collection of data related to a subroutine call that has not yet terminated with a return. Each frame is pushed onto a stack data structure.

1.1.16 Recursive Techniques

Recursive Technique: A subroutine that refers to itself in its definition.

1.2 Programming Paradigms

1.2.1 Programming Paradigms

Programming Paradigms: A style of computation and programming, chosen according to the problem at hand.

1.2.2 Structured Programming

Hierarchy Charts: A visual representation of a system design by breaking the most important elements into smaller sub tasks which can be individually coded (a top-down structure).



Structured Approach: A programming approach that relies on four basic structures to develop a program: assignment, sequence, selection and iteration.

1.2.3 Object-Oriented Programming

Aggregation: The creation of an object consisting of further objects such that these further objects continue to exist even if the initial object is deleted.

Class Diagrams: A visual representation of the relationships between classes and subclasses.

Classes: A template defining the attributes and methods that can be used to create a type of data known as an object.

Composition: The creation of an object consisting of further objects such that these further objects cease to exist if the initial object is deleted.

Encapsulation: A method of maintaining data integrity by only allowing class methods to access data in an object's attributes.

Inheritance: The concept of subclasses inheriting the methods and attributes of its parent class (a.k.a. its super class).

Instantiation: The creation of an object from a class.

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.

Objects: An instance of a class. The behaviour of this data item depends on how its attributes were defined.

Overriding: The redefinition of a method in a subclass, replacing the definition of the method with the same name in the parent class.

Polymorphism: Objects of different classes can use the same method to perform an action.

Definitions with a '+' taken from [AQA AS and A-level Computer Science specification version 1.5](#)

