

## Definitions and Concepts for OCR Computer Science A-level

### Component 1.2: Software and Software Development

---

#### **1.2.1 Systems Software**

**Basic Input Output System (BIOS):** A program that initialises and tests whether system hardware is functional and then loads the operating system from the hard disk into RAM when the computer is turned on.

**Device Drivers:** A program supplied with a peripheral device that allows the OS to control and communicate with the device.

**Distributed Operating Systems:** The operating system is spread over multiple computer servers on a network, acting as a single system to parallel process a job.

**Embedded Operating Systems:** A specialised operating system with limited resources and functionality, built in to control a single machine.

**First Come First Served:** A scheduling algorithm where processes are dealt with in the order they arrive (a queue).

**Intermediate Code:** Code partly translated between high-level and machine language produced by a compiler.

**Interrupts:** A signal from hardware, software or the clock to alert the CPU. If the interrupt is a higher priority than the current task, the current routine pauses and resumes after the interrupt is executed.

**Interrupt Service Routines (ISR):** If an interrupt is a higher priority than the current task, register contents are temporarily transferred onto the system stack at the end of the current FDE cycle and the interrupt is handled.

**Memory Management:** The efficient organisation and allocation of main memory to the programs in use

**Multi-level Feedback Queues:** A scheduling algorithm that uses multiple queues, each with a different priority. Jobs can be moved between queues.

**Multi-tasking Operating Systems:** An operating system capable of running multiple tasks simultaneously.

**Multi-User Operating Systems:** An operating system consisting of one mainframe computer

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/)



with multiple terminals that allow multiple users to access the computer's resources. Each user is given a time slice of the mainframe processor.

**Operating System:** A set of programs managing the operation of the computer that is loaded into RAM everytime the computer is turned on. It bridges the user to the hardware.

**Paging:** Partitioning memory into fixed sized physical divisions called pages. Processes in memory will be assigned an appropriate number of pages.

**Real Time Operating Systems:** An operating system where data is processed as it comes, with responses generated with a guaranteed timeframe.

**Round-Robin:** A scheduling algorithm where each process is given an equal time slice and is dealt with on a first in first out basis. If a process does not finish within a time slice it joins the end of the queue.

**Scheduling:** Allocating processor time to each application to ensure processor time is used as efficiently as possible when multitasking.

**Segmentation:** Partitioning memory into variable sized logical divisions called segments. A large program can be executed by consecutively running its segments.

**Shortest Job First:** A scheduling algorithm that picks the process with shortest estimated running time and runs it until it finishes.

**Shortest Remaining Time:** A scheduling algorithm that picks the process with shortest estimated time remaining to finish. If a process with a shorter time is added, the scheduler switches processes.

**Virtual Machines:** Any instance where software is used to take on the function of a machine, including intermediate code or running an operating system within another.

**Virtual Memory:** An allocated area of secondary storage where pages of inactive jobs are swapped into to free up enough RAM for the current job.

### **1.2.2 Applications Generation**

**Applications:** A program that can be run on a computer, allowing the user to carry out specific tasks.

**Assembler:** A translator in low level language, which converts assembly language into machine code.

**Closed Source:** Proprietary software sold with a licence and restrictions on how and how many users can use it. The source code is not available to users.

**Code Generation:** The third and final stage of compilation, where an equivalent machine code program is produced.



**Compilation:** The process of analysing high level language source code and converting it into machine code.

**Compilers:** A translator that converts high level language to machine code.

**Interpreters:** A translator which checks a source program for syntax errors line by line, translates it to machine code and executes the line.

**Lexical Analysis:** The first stage of compilation, where extra spaces and comments are removed from the source code and it is searched for simple errors. Keywords, constants and variables are replaced by tokens. Variable names are loaded into the symbol table.

**Libraries:** A collection of programs which are already compiled and can be loaded into a program and run whenever required.

**Linkers:** A program which places the appropriate machine addresses in the call and returns instructions of a compiled program so all the other required object code files and modules are linked together.

**Loaders:** A program that loads the executable object program and its associated libraries into memory before it is run.

**Open Source:** Software whose source code is freely available to view, redistribute or modify.

**Optimisation:** During code generation, the object code is made as efficient as possible by removing redundancies to produce code that gives the same result.

**Syntax Analysis:** The second stage of compilation, where statements, expressions and tokens are checked for syntax errors using syntax diagrams.

**Translator:** A program which converts code from one computer language to another.

**Utilities:** System software with a specific purpose usually related to maintenance such as optimising the performance of the computer, diagnosing issues, backing up files, setting up firewalls etc.

### **1.2.3 Software Development**

**Agile Methodologies:** An iterative process that produces incremental prototypes of software over short, fast-paced sprints. Each prototype is user tested and any feedback and change in requirements will be accounted for in future sprints.

**Extreme Programming:** A type of Agile Methodology that is more responsive to changing user needs with short development cycles and very frequent software releases. Checkpoints are incorporated to change or add new user requirements.

**Rapid Application Development :** A method capable of speedy responses to changes in



technologies and user requirements through repeated prototyping, continual evaluations, and strict time limits. The user will quickly receive a reduced-functionality mock up of the program.

**Spiral Model:** An iterative version of the Waterfall model where stages are refined and repeated until the final product is complete. The first cycle works towards an initial prototype, and each successive cycle produces a refined prototype.

**Waterfall Lifecycle:** Each stage of development is completed one at a time in a linear order. Results from a completed stage are input into the next. Any previous stage can be returned to in light of feedback, however the stages that follow need to be worked through again.

### **1.2.4 Types of Programming Languages**

**Assembly Language:** A low level language closely related to but more advanced than machine code. It uses descriptive names and mnemonics for instructions, and is related to the design of the computer.

**Attributes:** Data recorded as a variable associated with an object.

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

**Direct Addressing:** The simplest addressing mode in assembly language where the operand stores the memory address of the value to be operated on by the operator.

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

**Immediate Addressing:** An addressing mode in assembly language where the data in the address field is constant and the operand is the value to be operated on by the operator.

**Indexed Addressing:** An addressing mode in assembly language where the address of the operand required is calculated by adding a constant from the index register to the absolute address in the instruction.

**Indirect Addressing:** An addressing mode in assembly language where the operand stores the address of the location holding the memory address of the required data.

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

**Methods:** A program subroutine that represents an action an object can perform.

**Object-Oriented Languages:** Programming languages where the code is made of units called objects, which are instances of a class. Objects have their own attributes and behaviours and can interact with each other.



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

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

**Procedural Languages:** A high level language where statements are grouped into self-contained blocks known as functions and procedures. They have built in data types and data structures.

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

