

Section 20.1 - Programming Paradigms

Layer 6: High-Order Language

Syllabus Content Section 20: Further Programming

S20.1.1 Understanding what is meant by a programming paradigm

programming paradigm: A fundamental style of programming. Each paradigm will support a different way of thinking and problem solving. Paradigms are supported by programming language features

S20.1.2 Show understanding of the characteristics of a number of programming paradigms:

- Low-level Programming:
 - understanding of and ability to write low-level code that uses various addressing modes: immediate, direct, indirect, indexed and relative
 - Imperative (Procedural) programming:
 - Assumed knowledge and understanding of Structural Programming (see details in AS content section 11.3)
 - understanding of and ability to write imperative (procedural) programming code that uses variables, constructs, procedures and functions. See details in AS Content
 - Object-Oriented Programming (OOP):
 - understanding of the terminology associated with OOP (including objects, properties/attributes, methods, classes, inheritance, polymorphism, containment (aggregation), encapsulation, getters, setters, instances)
 - understanding of how to solve a problem by designing appropriate classes
 - understanding of and ability to write code that demonstrates the use of OOP
 - Declarative programming:
 - understanding of and ability to solve a problem by writing appropriate facts and rules based on supplied information
 - understanding of and ability to write code that can satisfy a goal using facts and rules
-

Low-level Programming

When you write with code that the CPU can directly understand

- Assembly Language
 - Machine Code / Binary
- Used for:

- Printer drivers
- Control Systems / Registers

Low level can communicate RAM. But we don't say communicate, we say address

Low Level has different ways to address RAM

1. Absolute / Direct
2. Indirect
3. Indexed
4. Immediate
5. Relative
6. Symbolic

Imperative (Procedural) programming

Your program runs in the same order that you typed it in.

It does not mean it uses an interpreter. Interpreter and compiler is how your code gets translated.

Procedural programming / Imperative programming just means whatever you typed first happens first.

After a while, they improved imperative programming by including procedures and functions

Object-Oriented Programming (OOP)

objects: an instance of a class

properties/attributes: the data items of a class (also sometimes referred to as fields)

methods: the subroutines of a class

Encapsulation: combining data and subroutines into a class and restricting external access to the data

classes: a type that combines a data structure with the methods that operate on the data structure

Inheritance: all attributes and methods of the base class are copied to the subclass

Polymorphism: the method behaves differently for different classes in the hierarchy

containment (aggregation): It just means when one class is made up from other classes

encapsulation: When you combine class attributes and class methods into one class

getters: a method to access its associated attribute

Containment: a relationship in which one class has a component that is of another class type

setters: a method to set the value of its associated attribute

instance: make a copy of an object

Declarative programming: