Struct and Class

Name: Mohammed Al-zubairi

What are the differences between Struct and Class?

Struct:

A struct is a way to group related variables together under one name. It is often used for simple data structures where the focus is mainly on storing data. By default, all members (variables and functions) inside a struct are public, meaning they can be accessed directly from outside the struct.

Class:

A class is a more advanced way to group related data and functions together. It is a foundational concept in Object-Oriented Programming (OOP). By default, all members inside a class are private, meaning they cannot be accessed directly from outside the class. You need to use functions to interact with the data.

Purpose and Usage:

Struct:

Structs are typically used for simple data structures where you just want to bundle some data together. They are straightforward and often used when you don't need to include much functionality beyond storing data. Common examples include representing simple entities like points, coordinates, or records in a program.

Class:

Classes are used when you need to combine data with behaviors (functions). They are more versatile and are used when you need to encapsulate both the data and the methods that operate on that data. Classes are central to Object-Oriented Programming, where you model real-world entities as objects that have attributes (data) and behaviors (functions).

Object-Oriented Features

Struct:

Structs are generally used when you need a simple container for data and do not need the advanced features of object-oriented programming.

Class:

Classes support all object-oriented features like inheritance (where a class can inherit properties from another class), polymorphism (where objects can be treated as instances of their parent class), and encapsulation. This makes classes suitable for building complex systems with well-organized code.