

MODULE: 4.1 (C++ Basic)

2) What is OOP? List OOP concepts

ANS:- OOP stands for Object-Oriented Programming.

It's a programming paradigm that revolves around the concept of "objects" and "class" which can contain data in the form of fields attributes or properties and code in the form of procedures methods or functions. Following are the concepts of OOP:

1)Class: It is a blueprint for creating objects. It defines the properties attributes and methods that objects of the class will have.

2)Object: It is an instance of a class which is a concrete entity that exists in memory and can be manipulated through methods defined in its class.

3)Encapsulation: It is the bundling of data (attributes) and methods (functions) that operate on the data into a single unit or class. It hides the internal state of an object from the outside world and only exposes the necessary functionality.

4)Abstraction: It is used to refers the concept of hiding complex implementation details and showing only the essential features of an object. It allows programmers to focus on what an object does rather than how it does it.

5)Inheritance: It is a mechanism where a subclass or derived class can inherit properties and behaviors from another class (superclass or base class). This promotes code reusability and establishes a hierarchical relationship between classes.

6)Polymorphism: It allows objects of different classes to be treated as objects of a common superclass. It enables a single interface to represent multiple underlying forms of data types and allows methods to behave differently based on the object they are called upon.

7)Message Passing: Objects communicate and interact with each other by sending messages. When one object invokes a method of another object, it sends a message to that object, instructing it to perform a certain operation.

3) What is the difference between OOP and POP?

ANS:- OOP (Object-Oriented Programming) and POP (Procedural-Oriented Programming) are two totally different programming paradigms, each with its own approach to structuring and organizing code. Following are some key differences between the two:

1)Approach to Problem Solving:-

OOP: OOP focuses on modeling real-world entities as objects, which have both data (attributes) and behavior (methods). It emphasizes the interaction and communication between these objects to solve problems.

POP: POP focuses on decomposing a problem into a set of procedures or functions, which operate on data. It emphasizes step-by-step procedure execution to achieve a task.

2)Data and Code Organization:-

OOP: In OOP, data and code are encapsulated within objects. Objects contain both data (attributes) and behavior (methods), which are tightly bound together. This promotes modularity, reusability, and maintainability.

POP: In POP, data and code are typically separated. Data is stored in variables, and functions operate on this data. Data and functions are not tightly coupled, leading to less encapsulation and more procedural code.

3)Abstraction:-

OOP: OOP emphasizes abstraction, where complex systems are represented by simplified models. Objects abstract real-world entities, and interactions between objects abstract real-world relationships and processes.

POP: POP also uses abstraction, but it tends to be less explicit. Abstraction in POP often involves breaking down a problem into smaller, more manageable procedures without necessarily representing real-world entities as objects.

4)Inheritance and Polymorphism:-

OOP: Inheritance and polymorphism are fundamental concepts in OOP. Inheritance allows the creation of hierarchical relationships between classes, promoting code reuse. Polymorphism enables objects of different classes to be treated uniformly.

POP: POP typically lacks explicit support for inheritance and polymorphism. While code reuse is possible through functions, it is not as structured or flexible as in OOP.

5)Code Reusability and Maintainability:-

OOP: OOP promotes code reusability and maintainability through features like encapsulation, inheritance, and polymorphism. Objects can be reused in different contexts, and changes to one part of the codebase often have minimal impact on other parts.

POP: Code reusability and maintainability in POP depend more on the programmer's discipline and organization. While functions can be reused, the lack of encapsulation and inheritance can make maintenance more challenging.