1) What is OOP? List OOP concepts?

ANS: OOP, or Object-Oriented Programming, is a programming paradigm that uses objects, which are instances of classes, to design and organize code. The primary goal of OOP is to promote modularity, reusability, and extensibility in software development. Here are some key concepts in OOP:

A) Class :

A blueprint or template that defines the attributes (data members) and behaviors (methods) common to all objects of a certain kind.

B) Object:

An instance of a class. Objects are created based on the class definition and represent individual entities with specific characteristics and behaviors.

C) Encapsulation:

The bundling of data and the methods that operate on that data into a single unit, called a class. It helps in hiding the internal details of an object and exposing only what is necessary.

D) Inheritance:

A mechanism that allows a class (subclass or derived class) to inherit properties and behaviors from another class (superclass or base class). This promotes code reuse and establishes an "is-a" relationship between classes.

E) Polymorphism:

The ability of objects of different classes to be treated as objects of a common base class. Polymorphism allows for flexibility and extensibility in code by enabling the use of a single interface to represent different types of objects.

F) Abstraction:

The process of simplifying complex systems by modeling classes based on the essential properties and behaviors they share. Abstraction involves focusing on the relevant aspects while ignoring unnecessary details.

G) Constructor:

A special method in a class that is automatically called when an object is created. It is used to initialize the object's state.

H) Destructor:

A special method in a class that is called when an object is about to be destroyed or deallocated. It is used to perform cleanup tasks.

I) **Method Overloading:**

The ability to define multiple methods in a class with the same name but different parameter lists. The correct method is chosen at compile-time or runtime based on the method signature.

J) **Method Overriding:**

The ability of a subclass to provide a specific implementation for a method that is already defined in its superclass. It allows for customization of behavior in derived classes.

K) **Association:**

A relationship between two or more classes, indicating that they are connected or interact with each other in some way. Associations can be one-to-one, one-to-many, or many-to-many.

L) **Composition:**

A strong form of association where one class contains an object of another class. It implies a "whole-part" relationship, and the lifetime of the contained object is dependent on the containing object.

M) **Aggregation:**

A weaker form of association where one class contains another class, but the contained class can exist independently. It implies a "has-a" relationship.

2) What is the difference between OOP and POP?

ANS: OOP (Object-Oriented Programming) centers on objects, encapsulation, inheritance, and polymorphism. POP (Procedural-Oriented Programming) focuses on procedures and sequential execution. OOP models real-world entities with modular, reusable code, while POP relies on procedures and functions for task-oriented programming without emphasizing object interactions.

3)