- **What is OOP? List OOP concepts?**

➢ Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.

➢ **<u>List OOP concepts</u>**

✓ Objects
✓ Classes
✓ Inheritance
✓ Data Abstraction
✓ Data Encapsulation
✓ Polymorphism
✓ Overloading
✓ Reusability

❖ Object

Object is the basic unit of object-oriented programming. Objects are identified by its unique name. An object represents a particular instance of a class. There can be more than one instance of a class. Each instance of a class can hold its own relevant data.

❖ Classes

Classes are data types based on which objects are created. Objects with similar properties and methods are grouped together to form a Class. Thus a Class represents a set of individual objects. Characteristics of an object are represented in a class as Properties. The actions that can be performed by objects become functions of the class and are referred to as Methods.

❖ Inheritance

Inheritance is the process of forming a new class from an existing class or base class. The base class is also known as parent class or super class. The new class that is formed is called derived class. Derived class is also known as a child class or sub class. Inheritance helps in reducing the overall code size of the program, which is an important concept in object-oriented programming.

❖ Data Abstraction

Data Abstraction increases the power of programming language by creating user defined data types. Data Abstraction also represents the needed information in the program without presenting the details.

❖ Data Encapsulation

Data Encapsulation combines data and functions into a single unit called class. When using Data Encapsulation, data is not accessed directly; it is only accessible through the functions present inside the class. Data Encapsulation enables the important concept of data hiding possible.

❖ Polymorphism

Polymorphism allows routines to use variables of different types at different times. An operator or function can be given different meanings or functions. Polymorphism refers to a single function or multi-functioning operator performing in different ways.

❖ Overloading

Overloading is one type of Polymorphism. It allows an object to have different meanings, depending on its context. When an existing operator or function begins to operate on new data type, or class, it is understood to be overloaded.

❖ Reusability

This term refers to the ability for multiple programmers to use the same written and debugged existing class of data. This is a time saving device and adds code efficiency to the language.

- **What is the difference between OOP and POP?**

Type	POP	OOP
Full Name	Procedure Oriented Programming	Object-Oriented Programming
Divided Into	In POP, the program is divided into small parts called functions.	In OOP, the program is divided into parts called objects.
Importance	In POP, functions and the order of operations to be performed take precedence over data.	Because it works in the actual world, data takes priority over procedures and functions in OOP.
Approach	POP follows the Top-Down approach.	OOP follows the Bottom-Up approach.
Access Specifies	POP does not have any access specified.	OOP has access specifies named Public, Private, Protected, etc.
Data Moving	In POP, Data can move freely from function to function in the system.	In OOP, objects can move and communicate with each other through member functions.
Expansion	To add new data and functions in POP is not so easy.	OOP provides an easy way to add new data and functions.
Data Access	Most functions in POP employ global data for sharing, which can be accessed freely from one function to the next.	Data in OOP cannot simply flow from one function to function; it can be kept public or private, allowing us to regulate data access.
Data Hiding	POP does not have any proper way for hiding data so it is less secure.	OOP provides Data Hiding so provides more security.
Overloading	In POP, Overloading is not possible.	In OOP, Overloading is possible in the form of Operator Overloading and Function Overloading.
Examples	Examples of POP are C, VB, FORTRAN, Pascal.	Examples of OOP are C++, JAVA, VB.NET, C#.NET.