**2). What is OOP? List OOP concepts.**

**Ans.**

* OOP stands for Object-oriented programming (OOP).
* OOP is a computer programming model that organizes software design around data, or object, rather than functions and logic.
* An object can be defined as a data field that has unique attributes and behavior.
* OOP focuses on the objects that developers want to manipulate rather than the logic required to manipulate them.
* This approach to programming is well-suited for programs that are large, complex and actively updated or maintained.
* This includes programs for manufacturing and design, as well as mobile application.

**OOP Concept:**

* polymorphism
* inheritance
* Dynamic Binding
* Data abstraction and Encapsulation
* class
* object

**3). What is the difference between OOP and POP?**

**Ans.**

|  |  |  |
| --- | --- | --- |
| **Features** | **OOP****(Object Oriented Programming)** | **POP****(Procedural Programming)** |
| Key Focuse | Objects And Their Interactions | Procedures And Function |
| Data Management | Data and Behavior are encapsulated in objects. | Data and Behavior are separate entities |
| Abstraction | Encourages the use of abstract classes and interfaces | Does not emphasize abstraction |
| Inheritance | Supports inheritance, allowing classes to inherit properties and methods | Does not support inheritance |
| Polymorphism | Allow objects of different types to be treated as the same type | Does not provide inherent polymorphism |
| Code Reusability | High level of code reusability thorough inheritance and composition | Relies on function and Subroutines for code Reusability |
| Code Organization | Follows a modular Approach with objects as self-contained modules | Relies on Functions and Procedures for Code Organization |
| Flexibility | Provides Flexibility through polymorphism and dynamic binding | Relies on structured programming offering less flexibility |
| Complexity Management | Encourages managing complexity through encapsulation and abstraction | Manages complexity through modular code organization and stepwise design |
| Real-world Modeling | Well-suited for modeling real-world entities and their interactions | May not align well with real-world modeling, focusing more on processes |