**OOPS Concept**

1. **What is OOP? List OOP concepts**

* OOP treats data as critical element in program development and does not allow it to flow freely around the system.
* It ties the data more closely to the function that operates on it.
* Object Oriented Programming allows decomposition of program into a number of entities called objects and then builds data and function around these objects.
* Characteristics of Object Oriented Programming:

❖ Emphasis on data rather than procedure.

❖ Program are divided into objects.

❖ Data is hidden and cannot be accessed by external functions.

❖ Objects may communicate with each other through

functions.

❖ Follows bottom up approach in program design.

* Some of the basic concepts of object oriented programming

are:

❖ Objects

❖ Classes

❖ Data abstraction and encapsulation

❖ Polymorphism

❖ Inheritance

❖ Dynamic Binding

1. **What is the difference between OOP and POP?**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **OOP** | **POP** |
| **Basic Definition** | OOP is object-oriented. | POP is structure or procedure-oriented. |
| **Program Division** | The program is divided into objects. | The program is divided into functions. |
| **Approach** | Bottom-Up approach | Top-down approach |
| **Data Control** | Data in each object is controlled on its own. | Every function has different data, so there’s no control over it. |
| **Entity Linkage** | Object functions are linked through message passing. | Parts of a program are linked through parameter passing. |
| **Expansion** | Adding new data and functions is easy. | Expanding data and function is not easy. |
| **Inheritance** | Inheritance is supported in three modes: public, private & protected. | Inheritance is not supported. |
| **Access control** | Access control is done with access modifiers. | No access modifiers supported. |
| **Data Hiding** | Data can be hidden using Encapsulation. | No data hiding. Data is accessible globally. |
| **Overloading or Polymorphism** | Overloading functions, constructors, and operators are done. | Overloading is not possible. |
| **Friend function** | Classes or functions can be linked using the keyword “friend, only in C++. | No friend function. |
| **Virtual classes or functions** | The virtual function appears during inheritance. | No virtual classes or functions. |
| **Code Reusability** | The existing code can be reused. | No code reusability. |
| **Problem Solving** | Used for solving big problems. | Not suitable for solving big problems. |
| **Example** | C++, JAVA, VB.NET, C#.NET. | C, VB, FORTRAN, Pascal |