

Module – 4

OOPS CONCEPT

2. What is OOP ? List OOP Concepts.

Ans.

Object-Oriented Programming (OOP) is a programming paradigm that uses objects to represent and manipulate data. The main goal of OOP is to bind data and the functions that operate on it together so that only the function can access the data. OOP allows objects to interact with each other using four basic principles :-

1. Abstraction
2. Encapsulation
3. Inheritance
4. Polymorphism

3. What is the difference between OOP and POP?

Ans.

Object-Oriented Programming (OOP)

- OOP treats data as a critical element in the program development and does not allow it to flow freely around the system.
- In OOP, the major emphasis is on data rather than procedure (function).
- It ties data more closely to the function that operate on it, and protects it from accidental modification from outside function.
- OOP allows decomposition of a problem into a number of entities called objects and then builds data and function around these objects.
- The data of an object can be accessed only by the function associated with that object. However, function of one object can access the function of other objects.
- C++, Java, Dot Net, Python etc are the example of Object oriented programming (OOP) language.

Procedural Oriented Programming (POP)

- In the procedure oriented approach, large programs are divided into smaller programs known as functions.
- In POP, a program is written as a sequence of procedures or function.
- In POP, each procedure (function) contains a series of instructions for performing a specific task.
- During the program execution each procedure (function) can be called by the other procedures.
- To call a procedure (function), we have to write function name only.
- While we concentrate onto the development of functions, we give very little attention to the data that are being used by various functions.

- In POP, the major emphasis is on procedure (function) and not on the data.
- In a multi-function program, many important data items are placed as global so that they may be accessed by all the functions. Each function may have its own local data.
- Global data are more vulnerable to an accidental change by a function. In a large program it is very difficult to identify what data is used by which function.
- Examples of procedural oriented programming language are COBOL, FORTRAN, PASCAL, C programming language etc.