**MODULE – 4**

**Q- 1 What is OOP ? List concept of OOP.**

* **OOP stands for object oriented programming.**
* **In this concept program is divided into objects.**
* **Data is hidden cannot be accessed by external functions.**
* **Object may communicate with each other through function.**

**Basic concept of OOP::**

**Abstraction**

**Class**

**Polymorphism**

**Encapsulation**

**Inheritance**

**Object**

1. **Class ::**

**Class is a collection of data member and member function (object).Class always declared with keyword class.**

**We can’t pass a value directly in class.**

**Class is declared at once. No memory is allocated for a class.**

1. **Object ::**

**Object is a instance of class. At a time of object creation we doesn’t required any keyword.**

**We can pass a value directly through the object.**

**One or more object created for a particular class.**

**3.Encapsulation ::**

**In normal terms , encapsulation is defined as wrapping up of data and information under a single unit. In object-oriented programming, Encapsulation is defined as binding together the data and the functions that manipulate them.**

**Encapsulation in C++**

Class

1. **Abstraction ::**

**Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.**

**5 Polymorphism ::**

**The word polymorphism means having many forms. In simple words we define polymorphism as the ability of message to be displayed in more one form. C++ supports operator overloading and function overloading.**

**6 .Inheritance ::**

**The capability of class to derived properties and characteristics from another class is called Inheritance. It is one of the most important features of OOP. It supports the concept of reusability.**

**EX. When we want to create a new class and there is already class that includes some of the code that we want, we can derive our new class from the existing class. By doing this we are reusing the fields and methods of the existing class.**