**Characteristics of Object-Oriented Languages**

The fundamental idea behind object-oriented system is to combine into a single unit both data and the functions that operate on that data. Such a unit is called an object.

Let’s briefly examine a few of the major elements of object-oriented languages in general, and C++ in particular.

**Objects**

Object is an identifiable entity with some characteristics and behaviour. It may represent a person, a

place, a bank account, a table of data or any item that the program must handle.

The term objects mean a combination of data and program that represent some real word

entity. For example: consider an example named Amit; Amit is 25 years old and his salary is 25000.

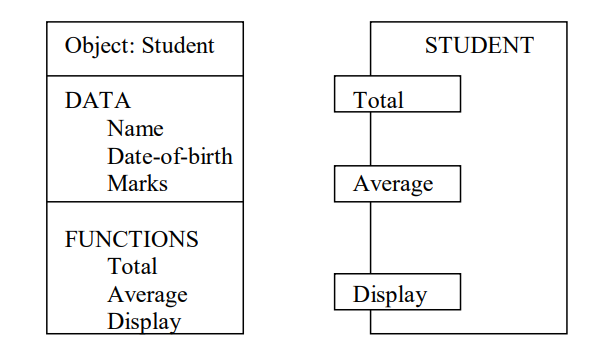
The Amit may be represented in a computer program as an object. The data part of the object would

be (name: Amit, age: 25, salary: 25000)

The program part of the object may be collection of programs (retrieve of data, change age,

change of salary). In general, even any user –defined type-such as employee may be used. In the

Amit object the name, age and salary are called attributes of the object.



**Class:**

A class is a template or blue print representing a group of objects that share common properties and relationships. For example, Car is an object and Automobiles is a class.

**Data abstraction:**

Abstraction refers to the act of representing essential features without including the back ground details or explanations. Example: You are driving a car. You only know the essential features to drive a car. Eg. gear handling, steering handling, use of clutch, accelerator, brakes etc. What is happening inside is hidden from you like wiring, motor working etc.

**Data encapsulation:**

The wrapping up of data and function into a single unit is known as encapsulation. In a company, there are so many departments like Sales, Accounts, Payroll, Purchase, Production etc. Suppose an employee from production department wants to know how much raw material has been purchased for the next month. The production department employee would not be allowed himself go through the purchase dept. data files. Rather he/she will have to issue a memo to the ‘purchase’ requesting for the required information. The data is not accessible to the outside world and only those functions which are wrapped in the class can access it. These functions provide the interface between the objects data and the program.

**Modularity:**

The act of partitioning a program into individual components is called is called modularity. For example, music system. The music system comprises of speakers, record-player, cd-player, tuner etc.

**Inheritance:**

Inheritance is the process by which objects of one class acquire the properties of another

class. In the concept of inheritance provides the idea of reusability. This mean that we can add

additional features to an existing class without modifying it. This is possible by designing a new class

will have the combined features of both the classes. The class ‘car’ inherits some of its properties from the class ‘Automobiles’ which inherits some of its properties from the class ‘Vehicles’.

**POLYMORPHISIM:**

Polymorphism means the ability to take more than one form. An operation may exhibit different

instance. The behaviour depends upon the type of data used in the operation. If you give 5+7, it results into 12, the sum of 5 and 7. And if you give ‘A’ + ‘B’, it results into ‘ABC’, the concatenated strings.