# Chapter 6 BASIC CONCEPTS OF OOP

#### **Objectives:**

- Provides an overview of object oriented programming
- To understand concept of objects, classes and other related terminologies
- > To highlight advantages of OOP
- To highlight limitations of OOP
- To identify the application areas



An object-oriented program consists of many well-encapsulated objects and interacting with each other by sending messages

#### **6.1 Introduction**

Object oriented programming is the principle of design and development of programs using modular approach. Object oriented programming approach provides advantages in creation and development of software for real life applications. The advantage is that small modules of program can be developed in shorter span of time and these modules can be shared by a number of applications. The basic element of object oriented programming is the data. The programs are built by combining data and functions that operate on the data. In this chapter we learn about the basic concepts, advantages and terminologies used in Object oriented programming. Some of the object oriented programming languages are C ++, Java, C # and so on.

The object oriented programming methods use data as the main element in the program. The data is tied to the function that operates on the data and the other functions cannot modify the data tied to a given function. Thus in object oriented programming, a problem is decomposed into a number of components called objects. An object is a collection of set of data known as member data and the functions that operate on these data a known as member functions or Methods. The member data are encapsulated in an object and then can be accessed or modified only by the member functions. An object can be accessed only if permitted by other member functions. Various objects of a program can interact with each other by sending messages.

Object oriented programming methods modularize a program by creating memory area for data and member functions (methods) together as a single entity. All objects are created according to the specifications of the entity defined.

Object is the basic unit of OOP. To design OO Model, first a set of classes are defined. A class is a Template from which objects are created. The Template of a class specifies the data, member functions and their attributes.



Figure 6.1 Organization of Data and functions in object oriented programming

#### 6.2 Basic concepts of object oriented programming

The following are the major characteristics of any object oriented programming language. They are



### 6.2.1 Objects

Objects are basic building blocks for designing programs. An object may represent a person, place or a table of data. An object is a collection of data members and associated member functions. Each object is identified by a unique name. Every object must be a member of a particular class.

Ex: Apple, orange, mango are the objects of class fruit.

Objects take up space in memory and have address associated with them. For example, structure variable in a C program.

At the time of execution of a program, the objects interact by sending messages to one another. The objects can interact with one another without having to know the details of data or functions within an object.

Object-1(Student) Object-2(Teacher) Member data Member data Name Name Subject Regno Data members Data members Course Course Member functions Member functions Getdata() Getdata() Member functions Member functions Display() Display()

Fig 1.2 Objects containing member data and member funtions

### 6.2.2 Classes

The objects can contain data and code to manipulate the data. The objects can be made user defined data types with the help of a class. Therefore objects



Fig 1.3 Classes

are variables of the type class. A class is a way of grouping objects having similar characteristics. Once a class is defined, any number of objects of that class are created.

For example, planets, sun, moon are members of class solar system.

Classes are user defined data types. A class can hold both data and functions.

### 6.2.3 Data abstraction

Data abstraction permits the user to use an object without knowing its internal working. Abstraction refers to the process of representing essential features without including background details or explanations. Classes use the concept of abstraction and are defined as a list of abstract attributes such as size, weight and cost, and functions to operate on these attributes.

#### 6.2.4 Data encapsulation



Fig 6.5 Encapsulation of data

### 6.2.5 Inheritance



Data encapsulation combines data and functions into a single unit called class. Data encapsulation will prevent direct access to data. The data can be accessed only through methods (function) present inside the class. The data cannot be modified by an external nonmember function of a class. Data encapsulation enables data hiding or information hiding.

> In OOP, the concept of inheritance provides the idea of reusability. This means that we can add additional features to an existing class without modifying it. Thus the process of forming a new class from an existing class is known as Inheritance. The objects of one class acquire the properties of another class through inheritance.

The existing class is known as base class. The new class is known as derived class.

The derived class shares some of the properties of the base class. Therefore a code from a base class can be reused by a derived class. In addition to this the new class may combine features from two different base classes too. In single inheritance, each subclass has only one superclass. In Multiple inheritance, each subclass has more than one super class.

#### 6.2.6 Overloading

Overloading allows objects to have different meaning depending upon context. There are 2 types of overloading namely

- 1. Operator overloading
- 2. Function overloading

When an existing operator operates on new data type, it is called operator overloading.

Function overloading means two or more functions have same name ,but differ in the number of arguments or data type of arguments. Therefore it is said that (function name) is overloaded. Function overloading therefore is the process of defining same function name to carry out similar types of activities with various data items.

#### **6.2.7** Polymorphism

Polymorphism is a feature of object oriented programming where a function can take multiple forms based on the type of arguments, number of arguments and data type of return value.

The ability of an operator and function to take multiple forms is known as polymorphism.

Example 1.2: Consider the addition operation. In addition of 2 numbers the result is the sum of 2 numbers.

In addition of 2 strings the operation is string concatenation. When an operator behaves differently based on operands, then it is said that operator is overloaded. Similarly when same function is used for multiple tasks in the same program by changing argument type and number, it is known as function overloading.

### **6.2.8 Dynamic Binding**

Binding is the process of connecting one program to another. Dynamic binding means code associated with a procedure call is known only at the time of program execution routine.

#### **6.2.9 Message Passing**

In OOP, processing is done by sending messages to objects. A message for an object is request for execution of procedure. The request will involve a procedure (function) in the receiving object that generates desired results. Message passing involves specifying the name of object, the name of the function (message) and the information to be sent.

# **6.3 Advantages of OOP over earlier programming methods**

- > The programs are modularized based on the principle of classes and objects.
- Linking code & object allows related objects to share common code. This reduces code duplication and code reusability.
- Data is encapsulated along with functions. Therefore external non- member function cannot access or modify data, thus providing data security.
- Easier to develop complex software, because complexity can be minimized through inheritance.
- The concept of data abstraction separates object specification and object implementation.
- Creation and implementation of OOP code is easy and reduces software development time.
- OOP can communicate through message passing which makes interface description with outside system very simple.

# **6.4 Limitations of OOP**

The main disadvantages of using Object oriented programming are:

- > OOP software is not having set standards.
- The adaptability of flow diagrams and object oriented programming using classes and objects is a complex process.
- > To convert a real world problem into an object oriented model is difficult.
- > The classes are overly generalized.

# 6.5 Applications of object oriented programming

Object oriented programming approach is an easier method to design and implement programs. The programs are easier to upgrade and modify. The standard class libraries can be used by the programmers so that development time is minimized. The graphical user interface design for windows operating system using object oriented programming is the most interesting feature of programming. The common application areas of Object oriented programming are:

- Computer graphic applications
- CAD/CAM software
- Object –oriented Database
- > User Interface design such as windows
- Real-time systems
- Simulation and Modeling
- > Artificial intelligence and expert systems

#### **Points to remember**

- Object oriented programming: Object oriented programming is a programming paradigm that uses "objects" to design applications and computer programs. The OOP uses several techniques such as inheritance, abstraction, modularity, polymorphism and encapsulation.
- Object: Object represents data and associated functions as a single unit.
- Class: A class is a way of grouping objects having similar characteristics.
- Abstraction: Abstraction refers to the representation of essential features of an object as a program object. An abstract class defines an interface, but does not provide implementation details.
- Encapsulation: It is a way of combining data and associated functions into a single unit. Encapsulation implements abstraction.
- Inheritance: It is the capability of a class to inherit the properties of another class. The class that inherits the properties from another class is known as derived or subclass. The class that provides its properties to subclass is known as base class.
- Polymorphism: It is ability of a function to have same name and multiple forms. The appropriate function is called automatically by the compiler depending on the number and type of arguments.
- Message passing: The processing of data in object oriented programming is carried out by sending messages to objects.

### **One mark questions:**

- 1. What is the fundamental idea of object oriented programming?
- 2. What is an object?
- 3. Define the term class.
- 4. Define the term data abstraction
- 5. What is encapsulation?
- 6. What is meant by function overloading?
- 7. Define polymorphism
- 8. What is inheritance?
- 9. What is a base class?
- 10. What is a derived class?
- 11. How are base class and derived class related?
- 12. Define the term data hiding

### **Two marks questions:**

- 1. What is the significance of classes in OOP?
- 2. What is the difference between program module and an object?
- 3. Mention different types of inheritance.

4. Mention any two advantages of object oriented programming over earlier programming methods.

#### **Three mark questions**

- 1. Briefly discuss the classes and objects.
- 2. Explain inheritance
- 3. Write short notes on polymorphism.
- 4. Mention any 4 high level languages that follow object oriented programming approach.

### Five marks answer questions

- 1. Write the differences between procedural programming and object oriented programming
- 2. Explain advantages OOPs
- 3. Write the disadvantages of object oriented programming
- 4. Write the real life applications of object oriented programming.