8
General Concept of OOP

8.1 Introduction

The object-oriented programming (OOP) is a different approach to programming. Object oriented technology supported by C++ is considered the latest technology in software development. It is regarded as the ultimate paradigm for the modelling of information, be that data or logic.

8.2 Objectives

After going through this lesson, you would be able to:

- learn the basic concepts used in OOP
- describe the various benefits provided by OOP
- explain the programming applications of OOP.

8.3 Object-Oriented Programming

The object-oriented programming is a different approach to programming. It has been created with a view to increase programmer's productivity by overcoming the weaknesses found in procedural programming approach. Over the years many object-oriented programming languages such as C++ and smalltalk have come up and are becoming quite popular in the market. The major need for developing such languages was to manage the ever-increasing size and complexity of programs.

8.4 Basic Concepts

The following are the basic concepts used in object-oriented programming.

- Objects
- Classes
- Data abstraction
- Modularity
- Inheritance
- Polymorphism

8.4.1 Objects

It can represent a person, a bank account or any item that a program can handle. When a program is executed, the objects interact by sending messages to one another. For example, if 'customer' and 'account' are two objects in a program, then the customer object may send message to account object requesting for a bank balance. Each object contains data and code to manipulate data. Objects can interact without having to know details of each other's data or code. It is sufficient to know the type of massage accepted and the type of response returned by the objects.

8.4.2 Classes

We have just mentioned that objects contain data and function or code to manipulate that data. The entire set of data and code of an object can be made a user-defined data type with the help of a class. In fact objects are variables of type class. Once a class has been defined, we can create any number of objects associated with that class. For example, mango, apple and orange are members of class fruit. If fruit has been defined as a class, then the statement fruit mango, will create an object mango belonging to the class fruit.

8.4.3 Data Abstraction

Abstraction refers to the act of representing essential features without including the background details. To understand this concept more clearly, take an example of 'switch board'. You only press particular switches as per your requirement. You need not know the internal working of these switches. What is happening inside is hidden from you. This is abstraction, where you only know the essential things to operate on switch board without knowing the background details of switch board.

8.4.4 Data Encapsulation

Encapsulation is the most basic concept of OOP. It is the way of combining both data and the functions that operate on that data under a single unit. The only way to access the data is provided by the functions (that are combined along with the data). These functions are considered as member functions in C++. It is not possible to access the data directly. If you want to reach the data item in an object, you call a member function in the object. It will read the data item and return the value to you. The data is hidden, so it is considered as safe and far away from accidental alternation. Data and its functions are said to be encapsulated into a single entity.

8.4.5 Modularity

The act of partitioning a program into individual components is called modularity. It gives the following benefits.

- It reduces its complexity to some extent.
- It creates a number of well-defined, documented boundaries within the program.

Module is a separate unit in itself. It can be compiled independently though it has links with other modules. Modules work quite closely in order to achieve the program's goal.

8.4.6 Inheritance

It is the capability to define a new class in terms of an existing class. An existing class is known as a base class and the new class is known as derived class. Number of examples can be given on this aspect. For example, a motor cycle is a class in itself. It is also a member of two wheelers class. Two wheelers class in turn is a member of automotive class as shown in Fig. 8.1. The automotive is an example of base class and two wheelers is its derived class. In simple words, we can say a motor cycle is a two wheeler automotive.

C++ supports such hierarchical classification of classes. The main benefit from inheritance is that we can build a generic base class, i.e., obtain a new class by adding some new features to an existing class and so on. Every new class defined in that way consists of features of both the classes. Inheritance allows existing classes to be adapted to new application without the need for modification.

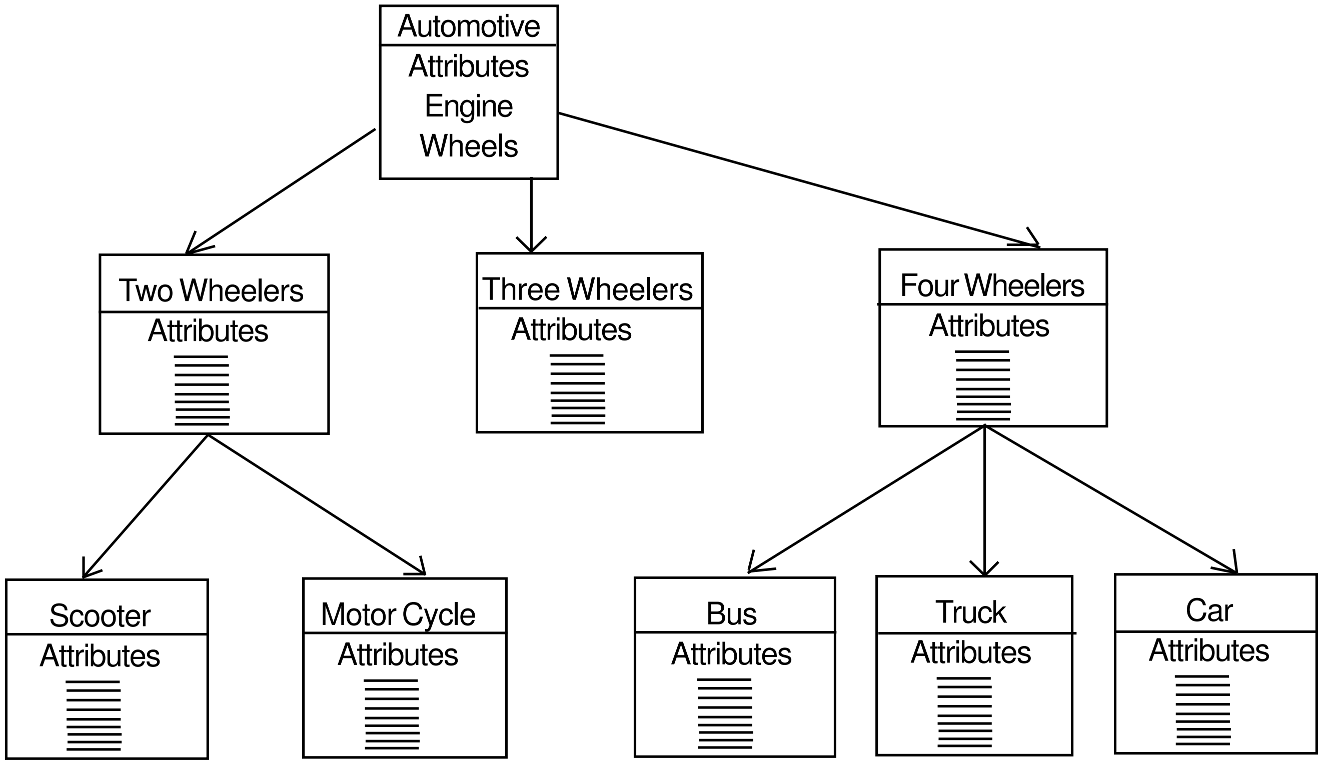

Fig. *8.1*

8.4.7 Polymorphism

Polymorphism is a key to the power of OOP. It is the concept that supports the capability of data to be processed in more than one form. For example, an operation may exhibit different behaviour in different instances. The behaviour depends upon the types of data used in the operation. Let us consider the operation of addition. For two numbers, the operation will generate a sum. If the operands are strings then the operation would produce a third string by concatenation.

In-Text Questions 8.1

1. State True (T) of False (F) in case of the following.

   (a) In procedure-oriented programming all data is shared by all functions.

   (b) One of the striking features of OOP is division of program into objects that represent real world entities.

(c) Wrapping up of data of different types into a single unit is known as encapsulation.

(d) Object oriented programs are executed much faster than conventional programs.

(e) Since C is a subset of C++, C programs will run under C++ compilers.

2. Why is data considered safe if encapsulated ?

3. Differentiate between data hiding and encapsulation.

8.5 Benefits of OOP

OOP provides lot of benefits to both the program designer and the user. Object-oriented approach helps in solving many problems related to software development and quality of software product. The new technology assures greater programmer productivity, better quality of software and lesser maintenance cost. The major benefits are :

- Software complexity can be easily managed
- Object-oriented systems can be easily upgraded
- It is quite easy to partition the work in a project based on objects.

8.6 Programming Applications of OOP

OOP has become one of the programming buzzwords today. There appears to be a great deal of excitement and interest among software programmers in using OOP. Applications of OOP are gaining importance in many areas. OOP has been extensively used in the development of windows and word based systems such as MS-Windows, x-Windows etc. The promising application areas of OOP are:

(i) Multiple data structure: This is an application where the same data structure is used many times. For example a window data structure is used multiple-times in a windowing system.

(ii) Data in multiple programs: This is an application where the same operations are performed on a data structure in different programs. For example, record validation in an accounting system.

The other application areas of OOP are parallel programming, simulation and modelling, AI and Expert systems, Neural Networks and CAD systems.

In-Text Questions 8.2

1. Fill in the blanks.

   (a) Small talk is an ....language.

   (b) An .... is a self contained unit of .... and function.

   (c) A .... can be used to create objects of its own type.

   (d) .... is a way of combining data with functions into an object.

   (e) A derived class can be derived from a .... class.

2. Mark True or False in front of the following :

   (a) C++ is an extension to C programming language.

   (b) Functions of one object cannot access the functions of other objects in C++

   (c) Inheritance is capability to define a new class in terms of an existing class.

   (d) Abstraction refers to the act of representing essential features without including the background details.

   (e) An object is a new data type.

3. Differentiate the following :

   (a) Procedural and object-oriented programming

   (b) An object and a class

8.7 What you have learnt

In this lesson you learnt about object-oriented technology, the latest technology in the software development. Various characteristics of OOP have been explained for the benefit of computer professionals. The promising application areas of OOP have also been described in brief.

8.8 Terminal Questions

1. What are the features of OOP ?

2. Explain the following terms briefly :

   (a) Data abstraction

   (b) Data encapsulation

   (c) Polymorphism

   (d) Inheritance

3. Describe briefly the various benefits of OOP?

8.9 Feedback to In-Text Questions

In-text Questions 8.1

1. (a) F (b) T

   (c) F (d) F (e) T

2. Encapsulation ensures that only authorised functions access that relevant data. The measure against unauthorised access ensures data safety.

3. Data hiding is a property when the internal data structure of an object is hidden from the rest of the program. On the other hand, data encapsulation is the mechanism by which data and its associated functions are bound together within an object definition in such a way that only relevant information is exposed and rest remains hidden. Thus data encapsulation implements data hiding.

In-text Questions 8.2

1. (a) OOP (b) Object, data (c) class

   (d) encapsulation (e) base

2. (a) T (b) F (c) T

   (d) T (e) F

3. (a) A program in a procedural language is a list of instructions which tell the

computer to do something. The focus is on the processing, the algorithm needed to perform the desired computation. In procedural paradigm, the emphasis is on doing things. The object-oriented approach views a problem in terms of objects involved rather than procedure for doing it.

(b) An object is an identifiable entity with some characteristics and behaviour. It represents an entity that can store data and its associated functions.

A class is a group of objects that share common properties and relationships.