

## EXPERIMENT NO. 01

DATE OF PERFORMANCE:

GRADE:

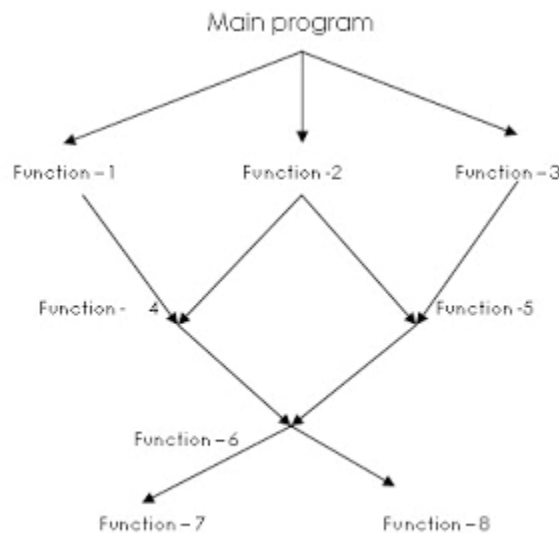
DATE OF ASSESSMENT:

SIGNATURE OF LECTURER/ TTA:

**AIM:** To Study Principles of Object Oriented Programming.

### **THEORY:**

Conventional programming, using high level languages such as COBOL and C, is commonly known as procedure oriented programming (POP). In the POP approach, the problem is viewed as a sequence of things to be done such as reading, calculating and printing. A number of functions are written to accomplish these tasks. The primary focus is on functions.



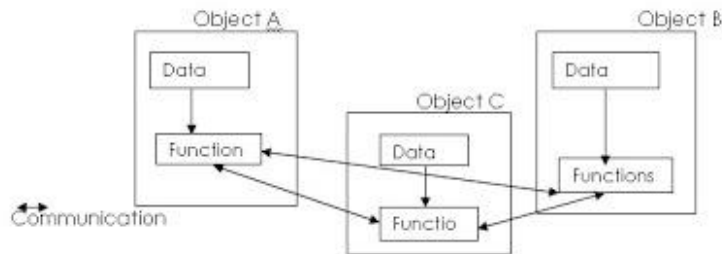
*Fig – 1.1 Typical structure of procedure oriented programs*

**Some characteristics exhibited by procedure-oriented programming are:**

- **Emphasis is on doing things (algorithms).**
- **Large programs are divided into smaller programs known as functions.**
- **Most of the functions share global data.**
- **Data move openly around system from function to function.**
- **Functions transform data from one form to another.**
- **Employs top-down approach in program design.**

# OBJECT - ORIENTED PROGRAMMING PARADIGM

The major motivating factor in the invention of object-oriented approach is to remove some of the flaws encountered in the procedural approach. OOP treats data as a critical element in the program development and does not allow it to flow freely around the system. It ties data more closely to the functions that operate on it, and protects it from accidental modification from outside functions. OOP allows decomposition of a problem into a number of entities called objects and then builds data and functions around these objects.



*Fig 1.2 The organization of data and functions in object – oriented programs*

**Some of the striking features of object – oriented programming are:**

- **Emphasis is on data rather than procedure.**
- **Programs are divided into what are known as objects.**
- **Data structures are designed such that they characterize the object.**
- **Functions that operate on the data of an object are tied together in the data structure.**
- **Data is hidden and cannot be accessed by functions.**
- **Objects may communicate with each other through functions.**
- **New data and functions can be easily added whenever necessary.**
- **Follows bottom-up approach in program design.**

## **Basic Concepts of Object – Oriented Programming**

**These include:**

- 1. Objects**
- 2. Classes**
- 3. Data abstraction and encapsulation**
- 4. Inheritance**
- 5. Polymorphism**
- 6. Dynamic binding**
- 7. Message passing**

## 1.OBJECTS:

Objects are the basic run time entities in an object – oriented system. They may represent a person, a place , a bank account, a table of data or any item that the program has to handle. They may also represent user – defined data such as vectors, time and lists. Programming problem is analyzed in terms of objects and the nature of communication between them.

When a program is executed, the objects interact by sending messages to each other. For example, If “customer” and “account” are two objects in a program, then the customer object may send a message to account object requesting for the bank balance.

## 2.CLASSES:

The entire set of data and code of an object can be made a user – defined data type with the help of a class. Infact objects are variables of the type class. Once a class has been defined, we can create any number of objects belonging to that class. Each object is associated with the data of type class with which they are created. A class is thus a collection of objects similar type. For example mango,apple, and orange are members of the class fruit. Classes are user defined data types and behave like the built in types of a programming language.

## 3.DATA ABSTRACTION AND ENCAPSULATION:

The wrapping up of data and functions into a single unit (called class) is known as encapsulation. Data encapsulation is the most striking feature of a class. 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 object’s data and the program. This insulation of the data from direct access by the program is called *data hiding* or *information hiding*.

*Abstraction* refers to the act of representing essential features without including the 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. They encapsulate all the essential properties of the objects that are to be created. The functions that operate on these data are sometimes called *methods* or *member functions*.

Since the classes use the concept of data abstraction, they are known as Abstract Data types (ADT).

## 4.INHERITANCE:

Inheritance is the process by which objects of one class acquire the properties of objects of another class. It supports the concept of *hierarchial classification*. For example, the bird ‘robin’ is a part of the class ‘flying bird’ which is again a part of the class ‘bird’.

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.

## 5. POLYMORPHISM:

Polymorphism is another important OOP concept. Polymorphism, a greek term means ability to take more than one form. An operation may exhibit different behaviors in different instances. The behavior depends upon the types of data used in the operation. For example, 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. The process of making an operator to exhibit different behaviors in different instances is known as *operator overloading*. A single function name can be used to handle different number and different types of arguments. This is something similar to a particular word having several different meanings depending on the context. Using a single function name to perform different types of tasks is known as *function overloading*.

## 6. DYNAMIC BINDING:

Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding (also known as late binding) means that the code associated with a given procedure call is not known until the time of the call at run-time. It is associated with polymorphism and inheritance.

## 7. MESSAGE PASSING:

An object-oriented program consists of a set of objects that communicate with each other. The process of programming in an object-oriented language, therefore involves the following basic steps:

- 1) Creating classes that define objects and their behavior,
- 2) Creating objects from class definitions, and
- 3) Establishing communication among objects.

Objects communicate with one another by sending and receiving information much the same way as people pass messages to one another.



## **BENEFITS OF OOPS:**

- Through inheritance, we can eliminate redundant code and extend the use of exiting classes.
- We can build programs from the standard working modules that communicate with one another, rather than having to start writing the code from scratch. This leads to saving of development time and higher productivity.
- The principle of data hiding helps the programmer to build secure programs that cannot be invaded by code in other parts of the program.
- It is possible to have multiple instances of an object to co-exist without any interference.
- It is possible to map objects in the problem domain to those in the program.
- It is easy to partition the work in a project based on objects.
- It is easy to partition the work in a project based on objects.
- The data-centered design approach enables us to capture more details of a model on implementable form.
- Object oriented systems can be easily upgraded from small to large systems.
- Message passing techniques for communication between objects makes the interface descriptions with external systems much simpler.
- Software complexity can be easily managed.

## **APPLICATIONS OF OOPS:**

**The promising areas for application of OOP include:**

- Real-time systems
- Simulation and modeling
- Object oriented databases
- Hypertext, hypermedia and expert text.
- AI and expert systems
- Neural networks and parallel programming
- Decision support and office automation systems
- CIM/CAM/CAD systems.