

Object Oriented Programming With C++

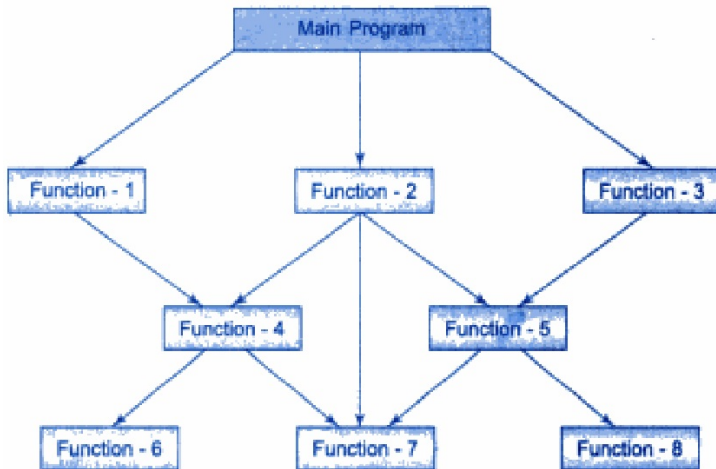
Dr. Monoj Kumar Muchahari

- 1 Introduction
 - Basic Concepts OOP

- 1 Introduction
 - Basic Concepts OOP

- Conventional programming, such as COBOL, FORTRAN and C, is commonly known as procedure-oriented programming (POP).
- In the procedure-oriented 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.
- The technique of hierarchical decomposition has been used to specify the tasks to be completed for solving a problem.

Procedure-Oriented Programming (POP) (contd.)

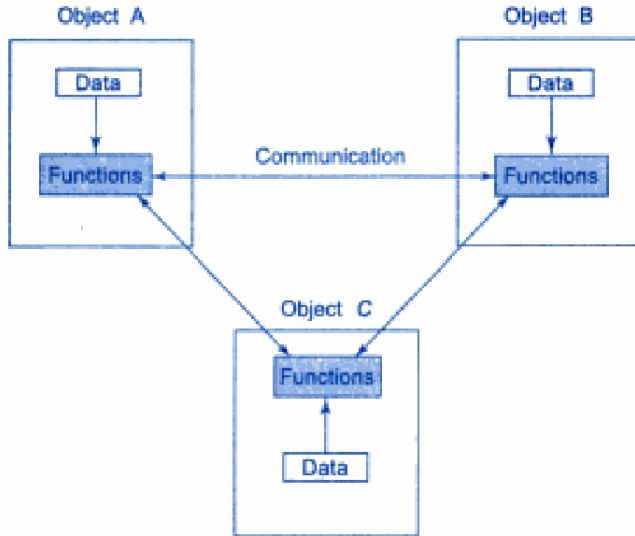


- While developers concentrate on the development of functions, very little attention is given to the data that are being used by various functions.
- In a multi-function program, many important data items are placed as global so that they may be accessed by all the functions.
- Global data are more vulnerable to an inadvertent change by a function.
- Another serious drawback with the procedural approach is that it does not model real world problems very well.

- Disadvantages of POP:
 - Limited code reusability
 - Difficulty in handling larger code complexity
 - Limited Real-World Modeling
 - Limited Encapsulation
 - Lack of Parallelism

- 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.

Object-Oriented Programming (contd.)



- Some of the **striking features of OOP** 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 objects.
 - Functions that operate on the data of an object are tied together in the data structure.
 - Data is hidden and cannot be accessed by external 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.

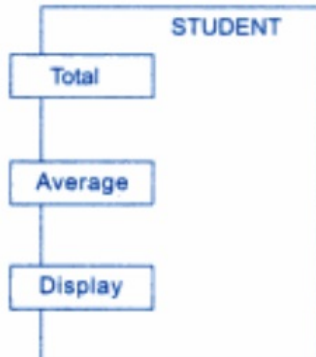
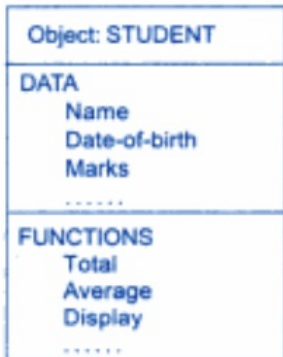
- Definition: *“object-oriented programming is an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand.”*
- Thus, an object is considered to be a partitioned area of computer memory that stores data and set of operations that can access that data.
- Since the memory partitions are independent, the objects can be used in a variety of different programs without modifications.

- 1 Introduction
 - Basic Concepts OOP

- The concepts used extensively in object-oriented programming include:
 - Objects
 - Classes
 - Data abstraction and encapsulation
 - Inheritance
 - Polymorphism
 - Dynamic binding
 - Message passing

- 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.
- Program objects should be chosen such that they match closely with the real-world objects.
- When a program is executed, the objects interact by sending messages to one another.

Objects (contd.)



- Objects contain data, and 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.
- 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 of similar type.
- For example, mango, apple and orange are members of the class fruit.

- The wrapping up of data and functions into a single unit (called class) is known as encapsulation.
- 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 **attributes** are sometimes called data members because they hold information.
- 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).**