

Chapter 1

Introduction

Features of C++

Some of the features of C++ are as follows

1. Stress is on the data rather than procedure.
2. Programs are divided into Objects.
3. Data is hidden & can not be accessed by external functions.
4. New data & function can be easily added whenever necessary.

Basic Concept of OOPS

Object-oriented remains a term which is interpreted differently by different people. It is therefore necessary to understand some of the concepts used extensively in object-oriented programming.

1) Object

Objects are the basic run time entities in OOP. An object may be a person, a place, a bank account etc. In programming the objects should be selected such that they should match with the real world objects. Each object contains data and the functions to manipulate the data. Object can interact without having to know details of each other's data or code. It is sufficient to know the type of message accepted and the type of response returned by the objects.

2) Class

It is a facility provided by OOP, which allows the user to combine the data and the functions in a single unit. The entire set of data and functions can be made user defined data type with the help of a class. Once a class has been define, we can create any no. of variables belonging to that class. The variables of the class are known as objects. 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. Classes are user-defined data types and behave like the built-in types of a programming language.

3) Data Encapsulation

The combining of data and functions into a single unit is known as Encapsulation. This facility is provided by the concept of class. Data encapsulation is the most striking feature of OOP. The data is not accessible to the outside functions and the functions which are declared in the class can only access the data. The encapsulation of data from direct access by the program is called data hiding.

4) Data Abstraction

Abstraction refers to the act of representing essential features without including the background detail 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. Since the classes use the concept of data abstraction, they are known as Abstract Data Types.

5) Inheritance

Inheritance is the process by which objects of one class acquires the properties of objects of another class. 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. This is possible by deriving a new class from the existing class. The new class will have the combine features of both the classes. Note that each sub-class defines only those features that are unique to it. Without the use of classification, each class would have to explicitly include all of its features.

6) Polymorphism

Polymorphism is another important concept of OOP. Polymorphism means the ability to take more than one form. 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 operand are strings, the operation would produce a third string by concatenation. This is something similar to a particular word having several different meaning depending on the context.

Polymorphism plays an important role in allowing objects having different internal structures to share the same external interface. This means that a general class of operations may be accessed in the same manner even though specific actions associated with each operation may differ. Polymorphism is extensively used in implementing inheritance.

e.g.

In C++, we can assign the same name to different functions in the program i.e. one name many forms. Such concept is known as Polymorphism.

7) Dynamic Binding

Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic 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. A function call associated with a polymorphism reference depends on the dynamic type of that reference.

8) Messages

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

1. Creating classes that define objects and their behaviour.
2. Creating objects from class definitions.
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. The concept of message passing makes it easier to talk about building systems that directly model or simulate their real-world counterparts.

Data Hiding

A property whereby the internal data structure of an object is hidden from the rest of the program. The data can be accessed only by the functions declared within class (of that object).

Reusability & Extensibility

A feature which is supported in object-oriented programming. This allows the reuse of existing classes without redefinition.

A feature that allows the extension of existing code. This allows the creation of new objects from the existing ones.
