HSSLIVE.IN

Chapter 2
Concepts of Object Oriented Programming

Programming Paradigm

It denotes the way in which a program is organised. Procedural paradigm and the object-oriented paradigm (OOP) are two important programming paradigms. C++ language supports both of these.

Procedural paradigm V/s OOP

Procedural paradigm	Object Oriented Paradigm
• Data is undervalued. • Procedure is given importance. • Creating new data types is difficult. • Poor real world modeling.	• Data is given importance. • Procedure is driven by data. • New data types and associated operations can easily be defined. • Easy to define real world scenarios.

Classes and Objects

A class is a prototype/blue print that defines data and functions common to all objects of a particular type. Object is an instance of a class with a unique set of data and functions as members. Class is an abstract data type and object is variable declared using a class. Student is a class with admission number, name and fee as data members. Anil may be an object of this class with values 1203, “Anil Kumar”, and 2000 for these data members. Setting these values to these data members is a function.

Bird is a class and Parrot is an object. Fruit is a class and Apple is an object. Complex Number is a class and 5+i2 is an object.

Class V/s Structure

Class	Structure
1. Contains data and functions as members. 2. Access specifiers private, public and protected are used for members. 3. By default, the members are private.	1. Contains only data as members. 2. No access specifiers are used. 3. By default, the members are public

Basic Concepts of OOP

1. Data abstraction: It refers to showing only the essential features of the application and hiding the details from outside world.
2. Data encapsulation: It binds the data and functions together and keeps them safe.
3. Polymorphism: The ability to process objects differently depending on their data type or class.
4. Inheritance: It is the process by which objects of one class acquire the properties and behaviour of another class.
5. Modularity: A concept through which a program is partitioned into small segments called modules.

Types of Polymorphism:

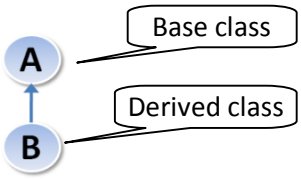
(1) Compile time (Static) polymorphism (or early binding): Polymorphism during and compilation. Function overloading and operator overloading are examples.

(2) Run time (Dynamic) polymorphism (or late binding): Polymorphism during run-time. It uses pointers. Virtual functions are examples.

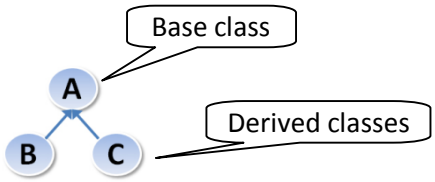
Function Overloading: Functions with the same name, but different signatures can act differently. Eg: The function prototypes `int area (int, int);` and `int area(int);` show that `area()` is an overloaded function.

Operator overloading: It is the process of giving new meaning to an existing C++ operator.

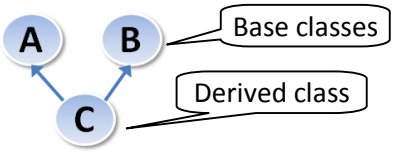
Different types of Inheritance



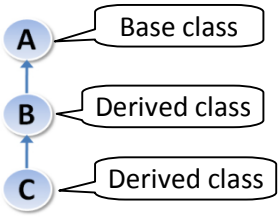
Single level Inheritance



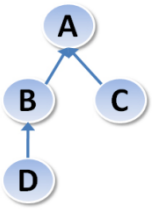
Heirarchical Inheritance



Multiple Inheritance



Multilevel Inheritance



Hybrid Inheritance

The class *from which* properties are derived is known as *base class*. The class *to which* the properties are derived is known as *derived class* or sub class.

Eg: Vehicle is a base class and Automobiles is a derived class.

Questions from Previous Years' Question Papers

1. Compare static and dynamic polymorphism. (3) (March 2017)
2. A program is implemented to find the area of a circle and areas of rectangle with two functions having same name but with different signature.
   (a) Name the concept. (1)
   (b) Explain this concept by writing the above program. (2) (SAY 2016)
3. Differentiate between data abstraction and data encapsulation. (3) (March 2017)
4. What is the difference between structure and class? (2) (SAY 2017)
5. Default access specifier is
   (a) private (b) public (c) protected (d) none (1) (SAY 2017)