3.3 Object Oriented Programming (C++)

Total marks: 75 (Semester end examination - 60, Internal assessment - 15)

Inheritence.

Date

An important feature of OOP is reusability.

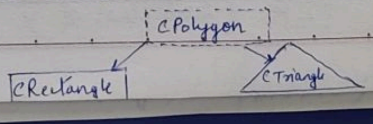
C++ supports the concept of reusability via inheritance or derivation. Inheritance is the capability of one class to inherit properties from an existing class. Inheritance allows to create classes which are derived from other classes, so that they automatically include some of its "parents" members, plus it's own.

The new derived classes are called sub classes and the old class in known as the base class or super class.

The derived class inherits the properties of the base class, including the member functions (methods).

For example : We are going to suppose that we want to declare a series of classes that describe polygons like our CRectangle, or like CTriangle. They have certain common properties, such as both can be described by means of only two sides : height and base.

This could be represented in the world of classes with a class CPolygon from which we would derive the two other ones : CRectángle and CTriangle.



3.3 Object Oriented Programming (C++)

otal marks: 75 (Semester end examination - 60, Internal assessment - 15)

Date

Different forms of Inheritance.

① Single Inheritance.

When a derived class inherits only from one base class, it is known as single inheritance,

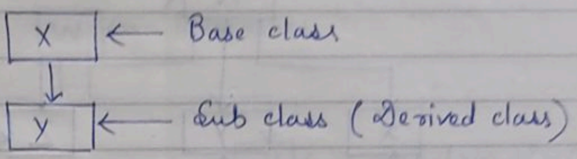


fig: 1 Single Inheritance.

② Multiple Inheritance.

when a derived class inherits from multiple base classes, it is known as multiple inheritance.

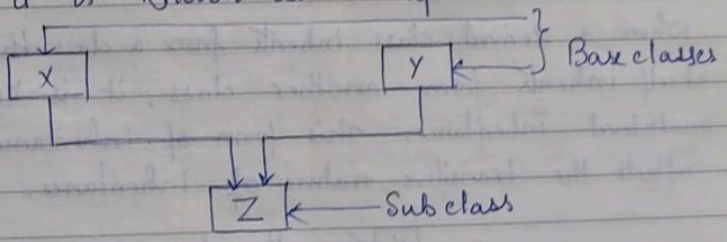


fig: 2 Multiple Inheritance.

3.

Total marks: 75 (Se

Date

(3) Hierarchical Inheritance.

When the propeoties of one class may be inherited by more than one class, it is known as hierarchical inheritance.

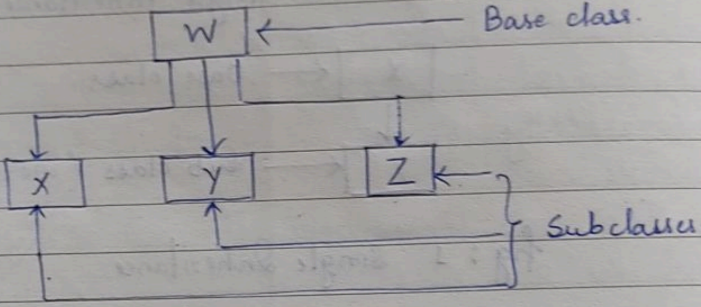


fig: 3. Hierarchical Inheritance.

(4) Multilevel Inheritance.

When a derived class inherits from a class that itself inherits from another class, it is known as multilevel inheritance. This form of inheritance reflects the transitive nature of inheritance.

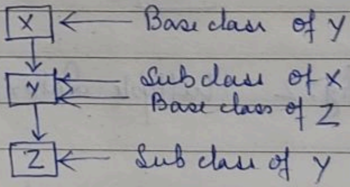


fig: 4 Multilevel Inheritance.

3.3 Object Oriented Programming (C++)

Total marks: 75 (Semester end examination - 60, Internal assessment - 15)

(5) Hybrid Inheritance.

Where a derived class inherits from multiple base classes and all of it's base classes inherit from a single base class, this form of inheritance is known as hybrid inheritance.

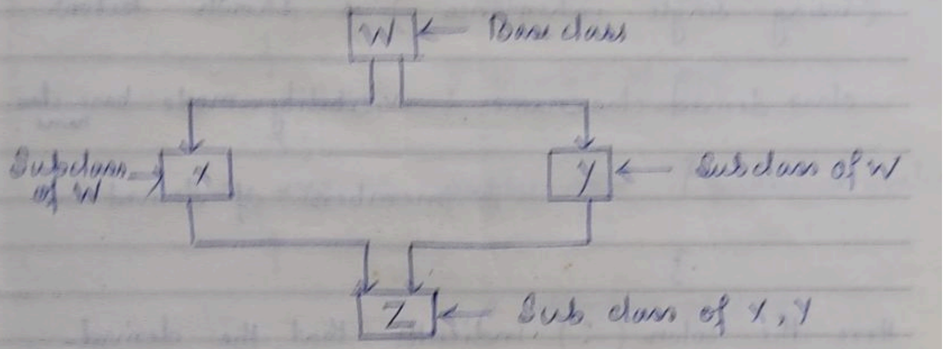


fig: 5. Hybrid Inheritance.

Total marks: 75

Defining derived class.

In C++, a derived class has to identify the class from which it is derived (i.e. it's base class) in addition to it's own details.

The syntax of defining a derived class (using single inheritance) is shown below:

```
class derived-class-name : visibility-mode base-class-name.
    {
        .....   // members of derived class.
        .....
    }
```

Here. the colon (:) indicates that the derived-class-name is derived from the base-class-name, the visibility-mode may be either private or public or protected.

The default visibility mode is private. The visibility mode controls the visibility and availability of the inherited base class members in the subclass (derived class).

3.3 Object Oriented Programming (C++)

otal marks: 75 (Semester end examination - 60, Internal assessment - 15)

for example :

```
class ABC  //  base class
    {
       .....   // member of ABC
    };
class  XYZ :   private ABC   // private derivation
    {
       .....   // member of XYZ
    };
class  XYZ  :  public  ABC  // public derivation
    {
       .....        // members of XYZ
    };
class  XYZ :  protected  ABC   // protected derivation
      {  ...        // member of XYZ .
      };
class  XYZ : ABC        // private derivation by
                           default .
      { ....        // members of XYZ .
      };
```