**Assignment No:** 3

**Problem Statement:** Implement C++/Java/Python program to create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called function get\_data() to initialize base class data members and another member function display\_area() to compute and display the area of figures. Make classes to suit their requirements. Using these three classes, design a program that will accept dimension of a triangle or a rectangle interactively, and display the area. Remember the two values given as input will be treated as lengths of two sides in the case of rectangles, and as base and height in the case of triangles, and used as follows:

Area of rectangle = x\*y

Area of triangle = ½\*x\*y;

**Aim of Assignment:** To implement inheritance and concept of virtual function to provide function overriding.

**Description:** Three classes named shape, rectangle and triangle are defined where shape is base class and rectangle and triangle are derived class which are publicly inherited. The base class has two virtual member functions called get\_data() and display\_area(). Functions are only declared in base class whereas defined in derived class. This is called as function overriding. The functions are defined as per the requirement in the base class. The base class pointer is used for calling the functions. The function call is decided at the run time , thus, implementing run time polymorphism.

**OOP Concept used:**

1. Inheritance: The capability of a class to derive properties and characteristics from another class is called **Inheritance**. Inheritance is one of the most important feature of Object Oriented Programming.  
   **Sub Class:** The class that inherits properties from another class is called Sub class or Derived Class.  
   **Super Class:**The class whose properties are inherited by sub class is called Base Class or Super class.
2. Function Overriding: If derived class defines same function as defined in its base class, it is known as function overriding in C++. It is used to achieve runtime polymorphism. It enables you to provide specific implementation of the function which is already provided by its base class.
3. Polymorphism: The word polymorphism means having many forms. Typically, polymorphism occurs when there is a hierarchy of classes and they are related by inheritance. C++ polymorphism means that a call to a member function will cause a different function to be executed depending on the type of object that invokes the function. It can be of two types – compile time polymorphism and run time polymorphism.
4. Virtual Function: A virtual function is a function in a base class that is declared using the keyword virtual. Defining in a base class a virtual function, with another version in a derived class, signals to the compiler that we don't want static linkage for this function. What we do want is the selection of the function to be called at any given point in the program to be based on the kind of object for which it is called. This sort of operation is referred to as dynamic linkage, or late binding.

**Conclusion:** The OOP concepts like inheritance, polymorphism and virtual function are implemented successfully.

**Program**:

#include<iostream>

using namespace std;

class shape

{

public:

double a,b;

virtual void get\_data()=0;

virtual void display\_area()=0;

};

class rectangle:public shape

{

public:

void get\_data()

{

cin>>a>>b;

}

void display\_area()

{

cout<<"\n Area of Rectangle is:"<<a\*b<<"\n";

}

};

class triangle:public shape

{

public:

void get\_data()

{

cin>>a>>b;

}

void display\_area()

{

cout<<"\n Area of Triangle is:"<<(a\*b)/2<<"\n";

}

};

int main()

{

int ch,ch1;

rectangle r1;

triangle t1;

do

{

cout<<"\n0.Exit\n1.Rectangle\n2.Triangle\n";

cout<<"\nEnter your choice\n";

cin>>ch;

switch(ch)

{

case 0:

break;

case 1:

cout<<"Enter length and breadth of rectangle\n";

r1.get\_data();

r1.display\_area();

break;

case 2:

cout<<"Enter base and heigth of triangle\n";

t1.get\_data();

t1.display\_area();

break;

case 3:

cout<<"Invalid choice plzz enter valid choice\n";

break;

}

}while(ch>0);

}



