Q1. Write a C++ Program to shown the concept of Virtual function also define a specific problem where we can use this concept for solving the specified problem using the Virtual function concept.

```
#include <iostream>
{
 public:
 virtual void display()
 {
  cout << "Base class is invoked"<<endl;
 }
};
class B:public A
{
 public:
 void display()
 {
  cout << "Derived Class is invoked"<<endl;
 }
};
int main()
{
 A* a;    //pointer of base class
 B b;     //object of derived class
 a = &b;
 a->display();   //Late Binding occurs
}
```

```
Derived Class is invoked
```

Q2. Write a C++ program to create an class 'Shape', which is the base class of classes 'Polygon' and 'Circle'. "Polygon" class is the base class from which a class Rectangular is derived. Now, create a virtual functions compute Area and compute Perimeter which are declared in the class Shape. These functions are actually defined in the derived classes "Rectangle" and "Circle". Define the classes "Shape", "Polygon", "Rectangle" and "Circle". Write a global function "ComputeAllArea" as

double computeAllArea(Shape*s[100], int numberofShapes). The above function computes the total area of all shapes object pointed to by s[0], s[1],....s[numberOfShapes].

```
#include <iostream>
using namespace std;

class Polygon {
  protected:
    int width, height;
  public:
    void set_values (int a, int b)
      { width=a; height=b; }
    virtual int area ()
      { return 0; }
};

class Rectangle: public Polygon {
  public:
    int area ()
      { return width * height; }
};

class Triangle: public Polygon {
  public:
    int area ()
      { return (width * height / 2); }
};

int main () {
  Rectangle rect;
  Triangle trgl;
  Polygon poly;
```

```
  Polygon * ppoly1 = &rect;
  Polygon * ppoly2 = &trgl;
  Polygon * ppoly3 = &poly;
  ppoly1->set_values (4,5);
  ppoly2->set_values (4,5);
  ppoly3->set_values (4,5);
  cout << ppoly1->area() << '\n';
  cout << ppoly2->area() << '\n';
  cout << ppoly3->area() << '\n';
  return 0;
}
```