Q1. Write program to generate following pattern

a)

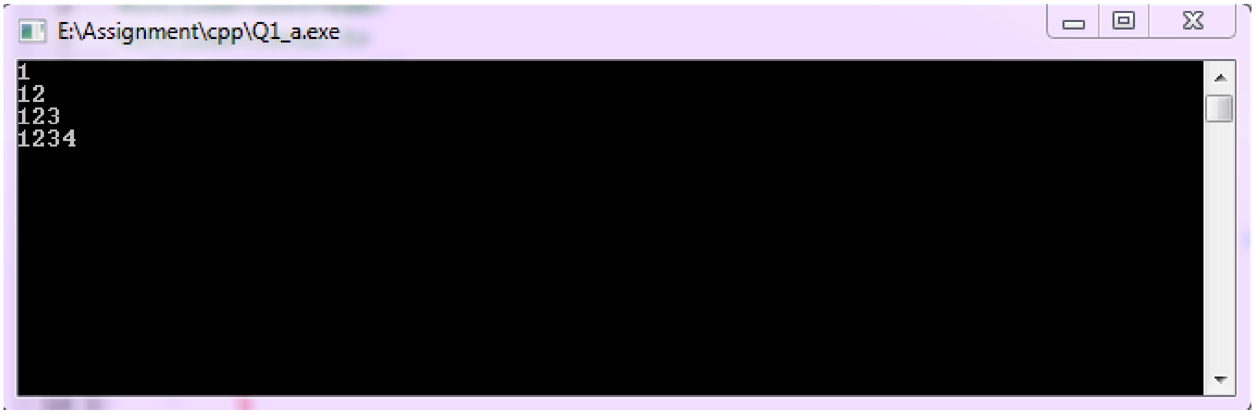
1
1 2
12 3
1 2 3 4

PROGRAM :

//Program to genarate Given pattern

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{	int total_rows = 4;
	for(int row = 1; row<=total_rows; row++) //Dynamic initilization of int row
	{
		for(int col=1; col<=row; col++)
		{
			cout<<col;
		}
		cout<<endl;
	}
	getch();
return 0;
}
```

OUTPUT:



```
1
12
123
1234
```

b)

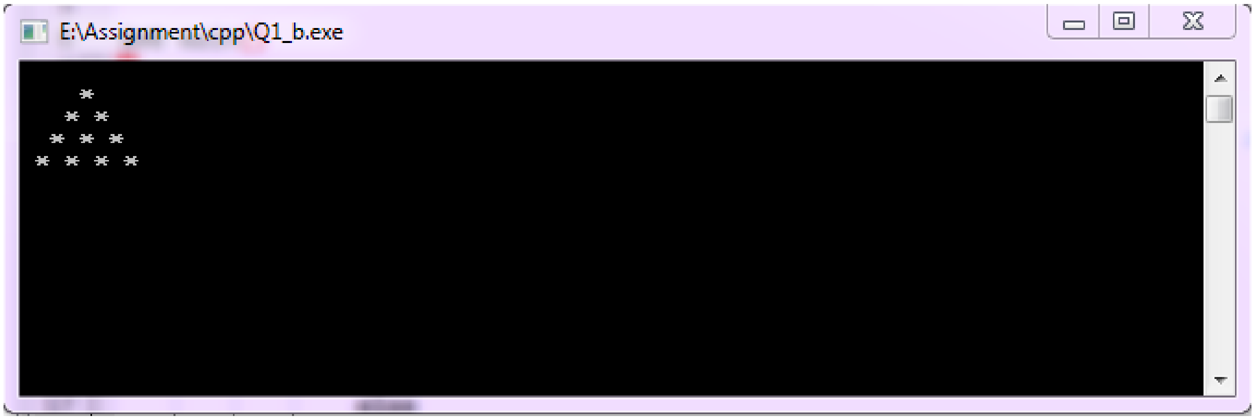
```
        *
      *  *
    *   *   *
  *   *   *   *
```

PROGRAM :

//Program to genarate pattern

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{           int total_rows = 4;
            for(int row = 0; row<=total_rows; row++) //Dynamic initilization of int row
            {
                        for(int col=0; col<=(total_rows * 2)-1; col++)
                        {
                                    if(col>total_rows-row && col < total_rows+row)
                                    {
                                                if((row+col)%2!=0)
                                                cout<<"*";
                                                else
                                                cout<<" ";
                                    }
                                    else
                                    cout<<" ";
                        }
                        cout<<endl;
            }
            getch();
return 0;
}
```

OUTPUT :



E:\Assignment\cpp\Q1_b.exe

```
    *
   * *
  * * *
 * * * *
```

Q2. WAP in C++ which uses functions to swap two integer & two float numbers by using reference variable.

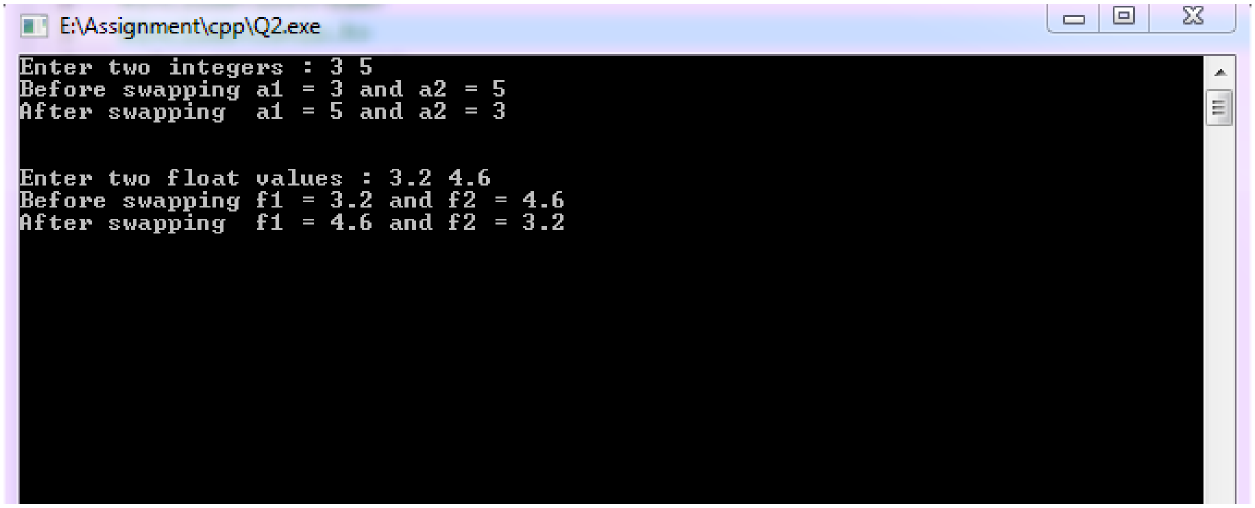
PROGRAM:

//Program to swap integer and float value using reference variable

```
#include<iostream>
#include<conio.h>
using namespace std;

void swap_int(int &,int &);
void swap_float(float &,float &);

                    //function declarations

void swap_int(int & x1,int & x2)                    //function definition
{
         int temp = x1;
         x1 = x2;
         x2 = temp;
}
void swap_float(float & x1,float & x2)             // x1 and x2 are reference variable
{
         float temp = x1;
         x1 = x2;
         x2 = temp;
}

int main()
{
         int a1 = 34;
         int a2 = 20;

         float f1 = 30.01;
         float f2 = 24.40;

         cout<<"Enter two integers : ";
         cin>>a1>>a2;
         cout<<"Before swapping a1 = "<<a1<<" and a2 = "<<a2<<endl;

         swap_int(a1,a2);                    //function calling
```

```
        cout<<"After swapping  a1 = "<<a1<<" and a2 = "<<a2<<endl;
        cout<<endl<<endl;
        cout<<"Enter two float values : ";
        cin>>f1>>f2;
        cout<<"Before swapping f1 = "<<f1<<" and f2 = "<<f2<<endl;

        swap_float(f1,f2);                    //function calling

        cout<<"After swapping  f1 = "<<f1<<" and f2 = "<<f2<<endl;
        getch();
        return 0;
}
```

OUTPUT:



```
Enter two integers : 3 5
Before swapping a1 = 3 and a2 = 5
After swapping  a1 = 5 and a2 = 3


Enter two float values : 3.2 4.6
Before swapping f1 = 3.2 and f2 = 4.6
After swapping  f1 = 4.6 and f2 = 3.2
```

Q3. Create a single program to perform following tasks without using library functions:

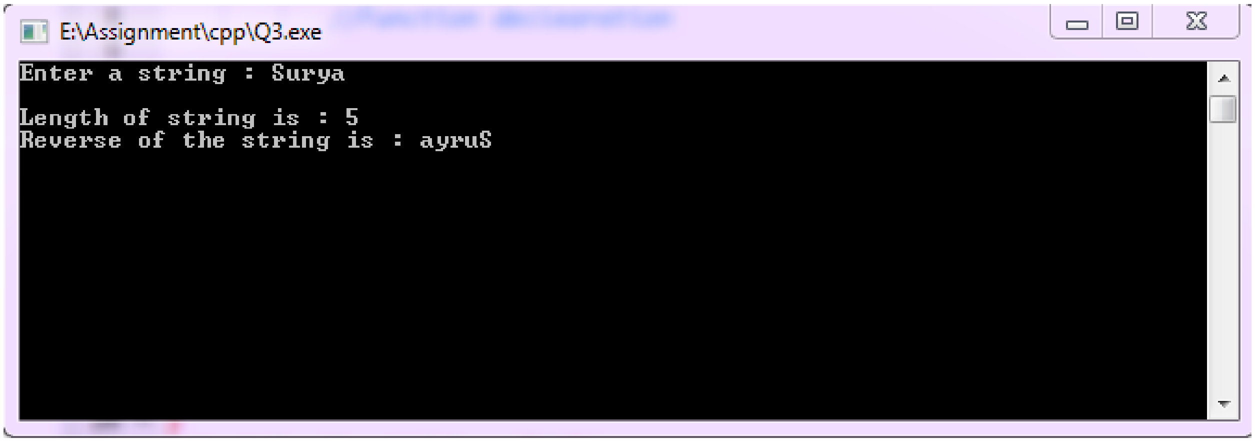
To reverse the string accepted as an argument.
To count the number of characters in string passed as argument in form of character array.

PROGRAM :

//Program to calculate length of string and reverse the string

```
#include<iostream>
#include<conio.h>
using namespace std;

int string_length(char);
void reverse_string(char);
                                        //Function declearation

void reverse_string(char rev[20],int length)          //function definition
{
      int i;
      cout<<"Reverse of the string is : ";
      for(i =length-1; i>=0; --i)
      {
            cout<<rev[i];
      }
      cout<<endl;

}

int string_length(char p[20])
{
  int count;
      for(count = 0; p[count]!= '\0'; ++count);
                                          //loop till getting null character
  return count;
}


int main()
{
      char str[20];
      cout<<"Enter a string : ";
      cin>>str;
      cout<<endl;
      int len = string_length(str);
                                    //storing length of string
            cout<<"Length of string is : "<<len<<endl;

            reverse_string(str,len);
                                            //passing length and string to the function
             getch();

            return 0;
}
```

OUTPUT:



```
Enter a string : Surya

Length of string is : 5
Reverse of the string is : ayruS
```

Q4. WAP in C++ to create a structure named complex having data member real and imag. Create member function add_complex which takes structure as an argument and return structure. Using function add two complex numbers.

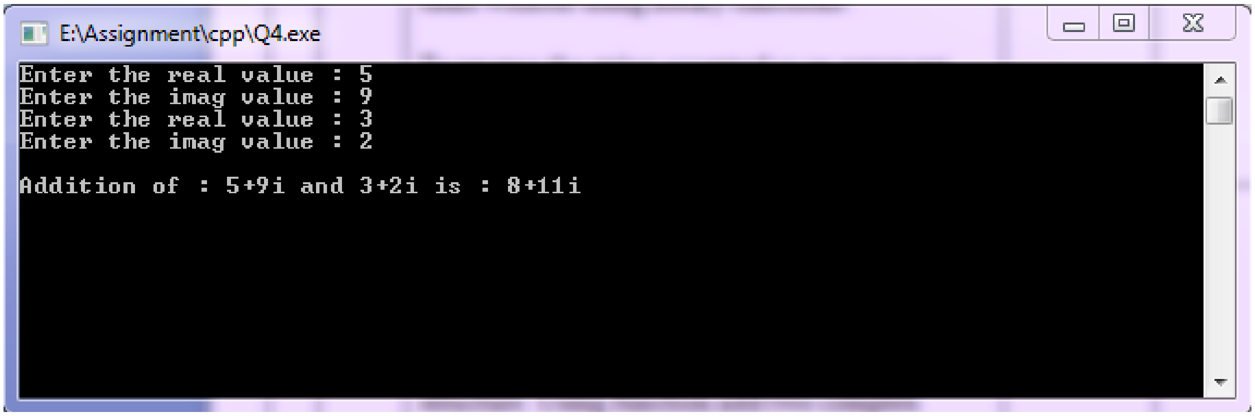
PROGRAM :

//Program to add to structures with trurn type structure(name complex)

```
#include<iostream>
#include<conio.h>
using namespace std;

struct complex                                          //Structure defined
{
  float real;
  float imag;                                          //data members

  void get_data()                                    //member function definition
  {
     cout<<"Enter the real value : ";
            cin>>real;
            cout<<"Enter the imag value : ";
            cin>>imag;
  }
  void display()                                      //member function definition
  {
           cout<<real<<"+"<<imag<<"i";
  }
};

complex add(complex,complex); //function  having  comlpex type arguments
complex add(complex c1,complex c2)
{
           complex c;
           c.real = c1.real + c2.real;  //adding real parts and storing in c.real
           c.imag = c1.imag + c2.imag;  //adding imag parts and storing in c.imag

           return c;  //returning a value of type complex
}

int main()
{
           complex comp1,comp2,add_comp;

           comp1.get_data();  //data members initilization
```

```
        comp2.get_data();
        add_comp = add(comp1,comp2);
                                //calling the function add and passing comp1 and comp2

        cout<<endl<<"Addition of : ";
        comp1.display();
        cout<<" and ";
        comp2.display();
        cout<<" is : ";
        add_comp.display();
        getch();
        return 0;
}
```

OUTPUT :



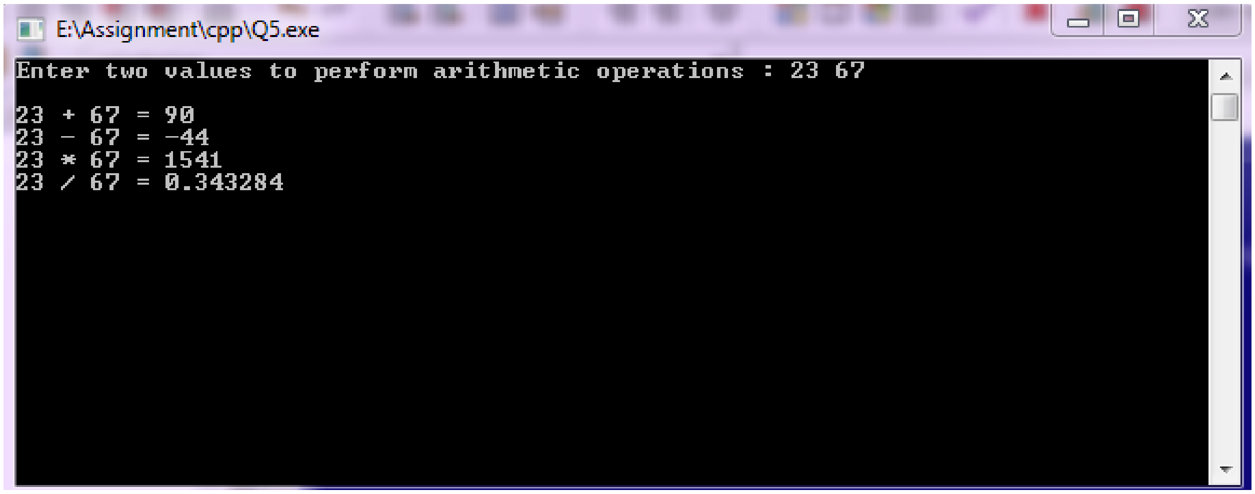
```
Enter the real value : 5
Enter the imag value : 9
Enter the real value : 3
Enter the imag value : 2

Addition of  : 5+9i and 3+2i is : 8+11i
```

Q.5 Write a program to perform arithmetic operations using inline function.

PROGRAM :

```
                //Program to perform arithmetic operations using inline function

#include<iostream>
#include<conio.h>
using namespace std;

inline void add(float a,float b)                                //inline function definition
{
        float c = a + b;
        cout<<a<<" + "<<b<<" = "<<c<<endl;
}
inline void subtract(float a,float b)                        //inline function definition
{
        float c = a - b;
        cout<<a<<" - "<<b<<" = "<<c<<endl;
}
inline void multi(float a,float b)                              //inline function definition
{
        float c = a * b;
        cout<<a<<" * "<<b<<" = "<<c<<endl;
}
inline void divide(float a,float b)                            //inline function definition
{
        float c = a / b;
   cout<<a<<" / "<<b<<" = "<<c<<endl;
}

int main()
{
        float n1,n2;
        cout<<"Enter two values to perform arithmetic operations : ";
        cin>>n1>>n2;
        cout<<endl;

        add(n1,n2);
        subtract(n1,n2);
   multi(n1,n2);
   divide(n1,n2);
                                                                 //inline function calling
   getch();
   return 0;
}
```

OUTPUT :



```
Enter two values to perform arithmetic operations : 23 67

23 + 67 = 90
23 - 67 = -44
23 * 67 = 1541
23 / 67 = 0.343284
```

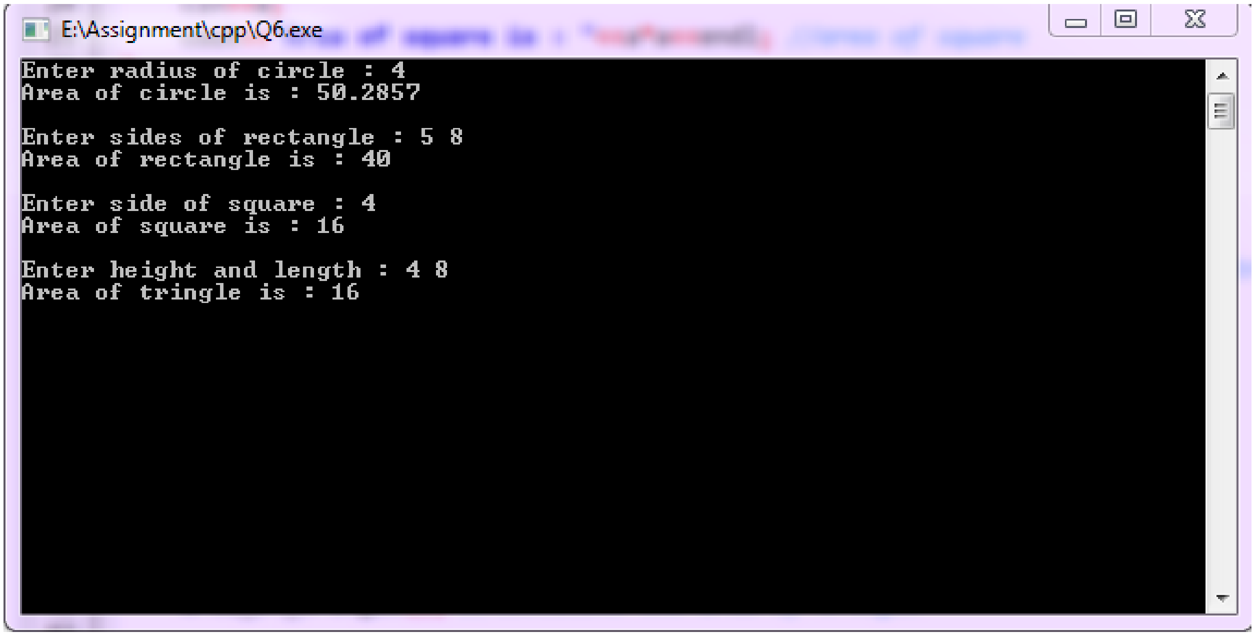
Q6. WAP in C++ to calclulate the area of circle, reractangle,square and triangle using inline function.

PROGRAM :

//Program to calculate the area of circle ,rectangle ,square and tringle using inline function

```
#include<iostream>

#include<conio.h>

using namespace std;

inline void area_of_circle()                                    //inline function definition

{

float radius;

  cout<<"Enter radius of circle : ";

  cin>>radius;

  cout<<"Area of circle is : "<<22/7.0 * radius * radius<<endl;          //area of circle

}

inline void area_of_rectangle()                                 //inline function definition

{

float a,b;

  cout<<"Enter sides of rectangle : ";

  cin>>a>>b;

  cout<<"Area of rectangle is : "<<a*b<<endl;                      //area of rectangle

}

inline void area_of_square()                               //inline function definition

{

float a;
cout<<"Enter side of square : ";
cin>>a;
cout<<"Area of square is : "<<a*a<<endl;                    //area of square
}
inline void area_of_tringle()                               //inline function definition
{
float height,length;
cout<<"Enter height and length : ";
cin>>height>>length;
cout<<"Area of tringle is : "<< 0.5 * height * length<<endl;          //area of tringle
}
int main()
{
area_of_circle();                         //calculate the area of circle
cout<<endl;
area_of_rectangle();                      //calculate the area of rectangle
cout<<endl;
area_of_square();                         //calculate the area of square
cout<<endl;
area_of_tringle();                        //calculate the area of tringle

getch();
return 0;
}
```

OUTPUT :



```
Enter radius of circle : 4
Area of circle is : 50.2857

Enter sides of rectangle : 5 8
Area of rectangle is : 40

Enter side of square : 4
Area of square is : 16

Enter height and length : 4 8
Area of tringle is : 16
```

Q7. WAP in c++ To count no. of vowels, consonants in each word of a sentence passed as argument in form of character array.

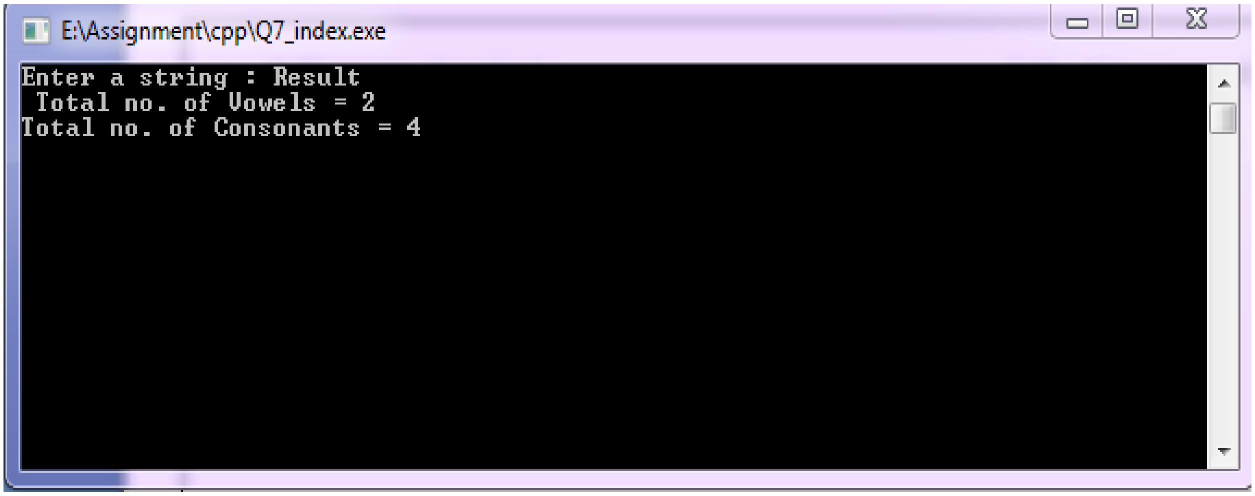
PROGRAM :

//Program to count no. of vowels and consonants in giver string

```
#include<iostream>
#include<conio.h>
#include<string.h>
using namespace std;

void count_vowel_conso(char);                    //fumction declaration
void count_vowel_conso(char s[20])               //function definition
{
  int i;
  bool lowercase,upercase;                        //bool type variable
  bool conso_check;
  int count_vowel=0;
  int count_conso=0;

  for(i=0; i<strlen(s); i++)
    {
   conso_check = ( ( ( s[i]>=97 ) && ( s[i]<=132 ) ) || ( ( s[i]>=65 ) && ( s[i]<=90 ) ) );
   lowercase = ( ( s[i] == 'a') ||( s[i] ==  'e')|| (s[i] ==  'i' )|| (s[i] ==  'o' ) || (s[i] ==  'u' ) );
   upercase = ( ( s[i] == 'A') ||( s[i] ==  'E')|| (s[i] ==  'I' )|| (s[i] ==  'O' ) || (s[i] ==  'U' ) );

                        if(lowercase || upercase )
                        count_vowel++;
                        else if(conso_check)
                        count_conso++;
     }

                        cout<<" Total no. of Vowels = "<<count_vowel<<endl;
                        cout<<"Total no. of Consonants = "<<count_conso<<endl;
}

int main()
{
            char str[20];
            cout<<"Enter a string : ";
            cin>>str;

            count_vowel_conso(str);
```

```
                                        //function calling and passing the string
        getch();
        return 0;
}
```

OUTPUT :



```
Enter a string : Result
 Total no. of Vowels = 2
Total no. of Consonants = 4
```

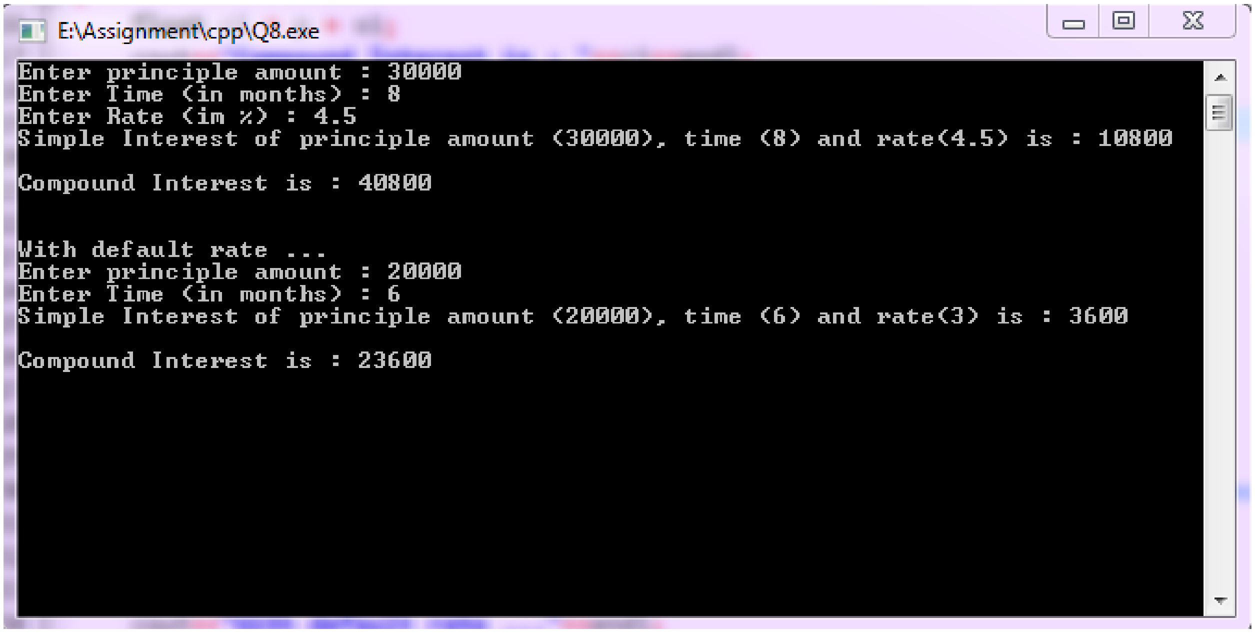
Q8. Write program in C++ to calculate simple interest and compound interest using default argument.

PROGRAM :

//Program to calculate simple interest and compound interest using default argument

```
#include<iostream>
#include<conio.h>
using namespace std;

float simple_interest(float,float,float);
void compound_interest(float,float);
                                                         //function declaration
float simple_interest(float p,float t, float r = 3)          //Default argument r (rate) is 3%
{
        float Si = (p * t * r ) / 100;
cout<<"Simple Interest of principle amount ("<<p<<"), time ("<<t<<") and rate("<<r<<") is :
"<<Si<<endl<<endl;

        return Si;
}
void compound_interest(float p,float si)
{
        float ci = p + si;
        cout<<"Compound Interest is : "<<ci<<endl;
}
int main()
{
        float princ,time,rate;

        cout<<"Enter principle amount : ";
        cin>>princ;
        cout<<"Enter Time (in months) : ";
        cin>>time;
        cout<<"Enter Rate (im %) : ";
        cin>>rate;

        float simp_int = simple_interest(princ,time,rate);
                                              //no. of actual and formal argument are same
                                              //Not a condition for default argument
        compound_interest(princ,simp_int);                //compount interest
        cout<<endl<<endl;
        cout<<"With default rate ..."<<endl;
        cout<<"Enter principle amount : ";
        cin>>princ;
        cout<<"Enter Time (in months) : ";
        cin>>time;

  simp_int = simple_interest(princ,time);
                              // no. of actual argument is less than
                             //formal argument( so default argument will be passed)
        compound_interest(princ,simp_int);

        getch();
        return 0;
}
```

OUTPUT :



```
Enter principle amount : 30000
Enter Time (in months) : 8
Enter Rate (im %) : 4.5
Simple Interest of principle amount (30000), time (8) and rate(4.5) is : 10800

Compound Interest is : 40800


With default rate ...
Enter principle amount : 20000
Enter Time (in months) : 6
Simple Interest of principle amount (20000), time (6) and rate(3) is : 3600

Compound Interest is : 23600
```

Q9. Create a class named calculate that uses overloaded function calclulate_area of circle, reractangle,square and triangle.

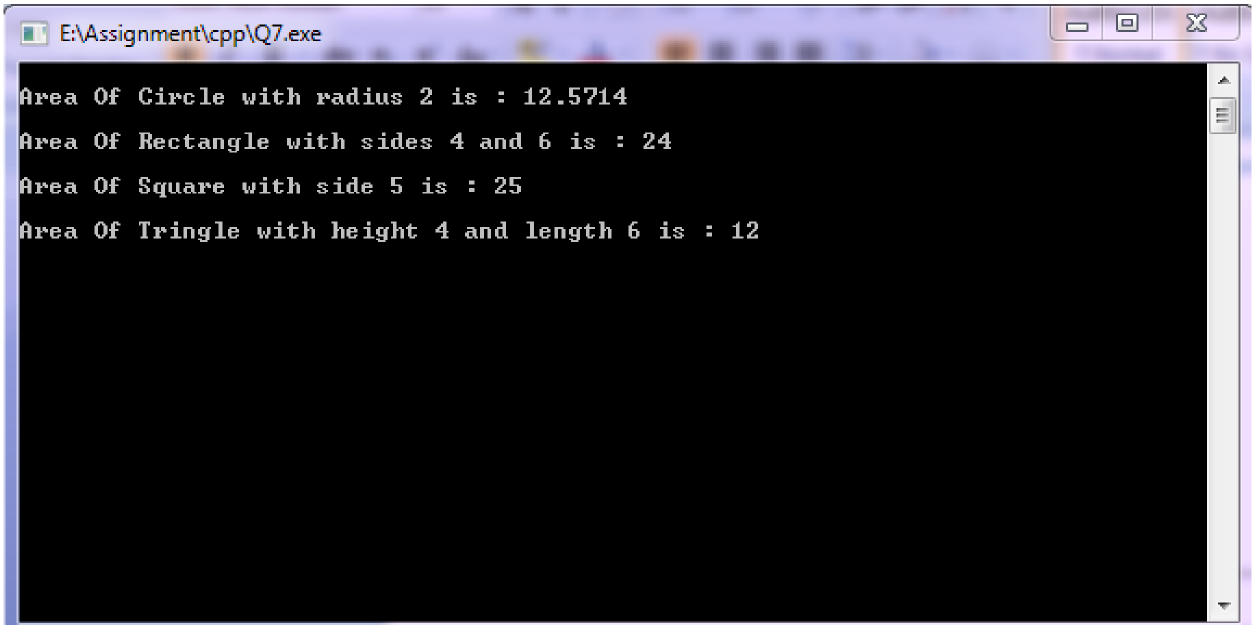
PROGRAM :

//Program to calculate area of circle,square,rectangle and tringle using function overloading

```
#include<iostream>

#include<conio.h>

using namespace std;


class calculate

{

        float area;

        public:

                void calculate_area(float);               // function overloading declaration

                void calculate_area(float,float);        // function overloading declaration

                void calculate_area(float,float,float);    // function overloading declaration

};

  void calculate :: calculate_area(float radius)                    //function defining outside of the class

        {

                area= 22/7.0 * radius * radius;

                cout<<"Area Of Circle with radius "<<radius<<" is : "<<area<<endl;

        }
        void calculate :: calculate_area(float length,float width)
        {
                area = length * width;
                if(length == width)
                cout<<"Area Of Square with side "<<length<<" is : "<<area<<endl;
                else
cout<<"Area Of Rectangle with sides "<<length<<" and "<<width<<" is : "<<area<<endl;
        }
        void calculate :: calculate_area(float half,float height,float length)
        {
                area = half * height * length;
cout<<"Area Of Tringle with height "<<height<<" and length "<<length<<" is : "<<area<<endl;
        }
int main()
{
        calculate obj;
        cout<<endl;
        obj.calculate_area(2);                          //calculate the area of circle
        cout<<endl;
        obj.calculate_area(4,6);                          //calculate the area of rectangle
        cout<<endl;
        obj.calculate_area(5,5);                          //calculate the area of square
        cout<<endl;
        obj.calculate_area(0.5,4,6);                      //calculate the area of tringle
```

```
  getch();

        return 0;     }
```

OUTPUT :



```
Area Of Circle with radius 2 is : 12.5714
Area Of Rectangle with sides 4 and 6 is : 24
Area Of Square with side 5 is : 25
Area Of Tringle with height 4 and length 6 is : 12
```

Q10. Create a class Student having data members to store roll number, name of student, name of three subjects, max marks, min marks, obtained marks. Declared an object of class student. Provide facilities to input data in data members and display result of student?

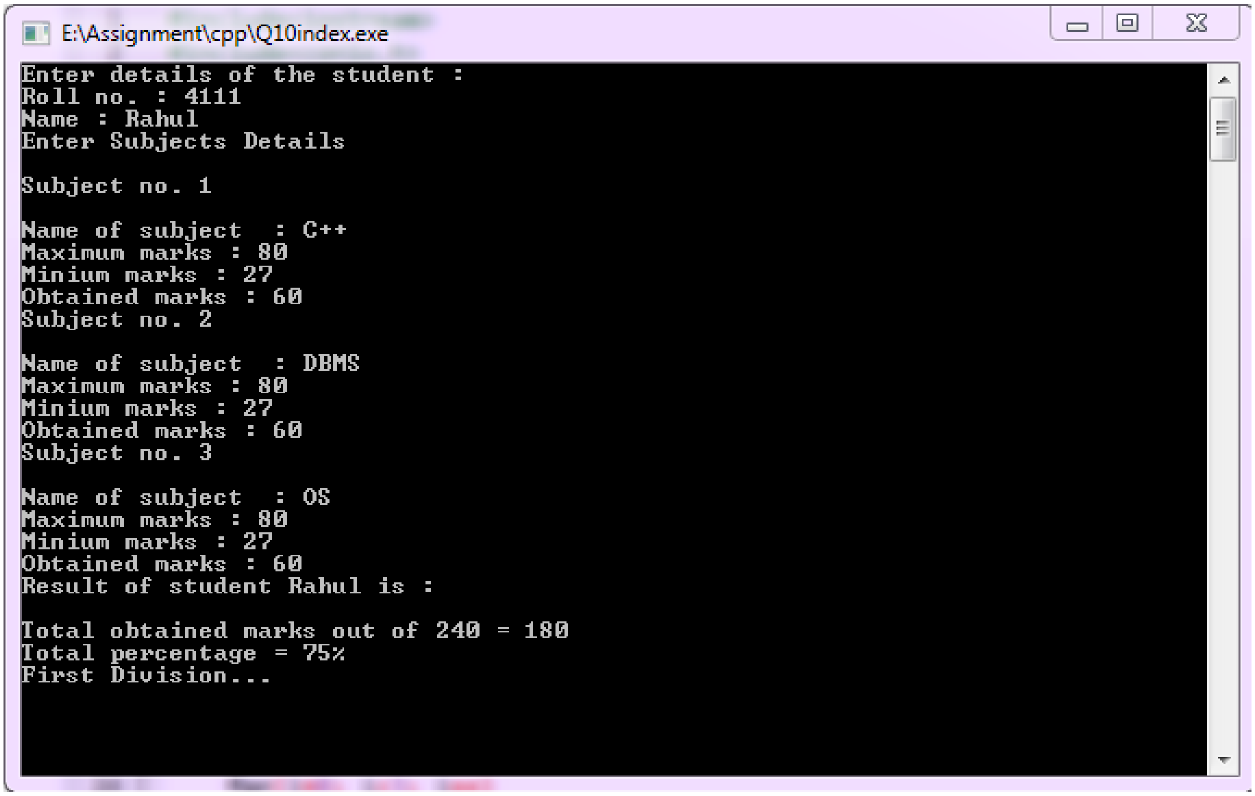
PROGRAM :

//Program to store data of class student and display result of student

```
#include<iostream>
#include<conio.h>
using namespace std;
class student
{        int rollno;
        char stu_name[20];
        char sub_name[3][20];
        float max_mark[3],min_mark[3],obt_mark[3];
                                                        //data members
          public:
                void input_data();
                void result();
                        //member function declaration
};

void student::input_data()                    //member function definition outside of class
{
        cout<<"Enter details of the student : "<<endl;
        cout<<"Roll no. : ";
        cin>>rollno;
        cout<<"Name : ";
        cin>>stu_name;
        cout<<"Enter Subjects Details "<<endl<<endl;
        int i;
        for(i=0; i<3; i++)
        {
          cout<<"Subject no. "<<i+1<<endl<<endl;
          cout<<"Name of subject  : ";
          cin>>sub_name[i];
          cout<<"Maximum marks : ";
          cin>>max_mark[i];
```

```
             cout<<"Minium marks : ";
             cin>>min_mark[i];
             cout<<"Obtained marks : ";
             cin>>obt_mark[i];
            }
      }

      void student::result()
      {
         cout<<"Result of student "<<stu_name<<" is : "<<endl<<endl;
         float total_max,total_obt;
         float per;
            int i;
            for(i=0; i<3; i++)
            {
                 total_max = total_max + max_mark[i];
                 total_obt = total_obt + obt_mark[i];
            }

            per = total_obt * 100 / total_max;
            cout<<"Total obtained marks out of "<<total_max<<" = "<<total_obt<<endl;
            cout<<"Total percentage = "<<per<<"%"<<endl;

            if(per >= 70)
            cout<<"First Division..."<<endl;
            else if(per < 70 && per >= 50)
            cout<<"Second Division..."<<endl;
            else if(per > 33 && 50 > per)
            cout<<"Third Division.."<<endl;
            else
            cout<<"Fail"<<endl;
      }


      int main()
      {
            student stu1;                              //Object created of type student
            stu1.input_data();
            stu1.result();
                                               //calling member function through object
            getch();
            return 0;
      }
```

OUTPUT :



```
Enter details of the student :
Roll no. : 4111
Name : Rahul
Enter Subjects Details

Subject no. 1

Name of subject   : C++
Maximum marks : 80
Minium marks : 27
Obtained marks : 60
Subject no. 2

Name of subject   : DBMS
Maximum marks : 80
Minium marks : 27
Obtained marks : 60
Subject no. 3

Name of subject   : OS
Maximum marks : 80
Minium marks : 27
Obtained marks : 60
Result of student Rahul is :

Total obtained marks out of 240 = 180
Total percentage = 75%
First Division...
```

Q11. Create a class student having data members to store rollno.,name of student, name of 3 subjects , max marks,min marks,obtain marks .use nesting of member function Declare an array of object to input data of 3 students. Provide facilities to display result of all students and to display result of specific student whose roll number is given ?

PROGRAM:

//Programn to store data of 3 students and display result of all and also display result of specific student

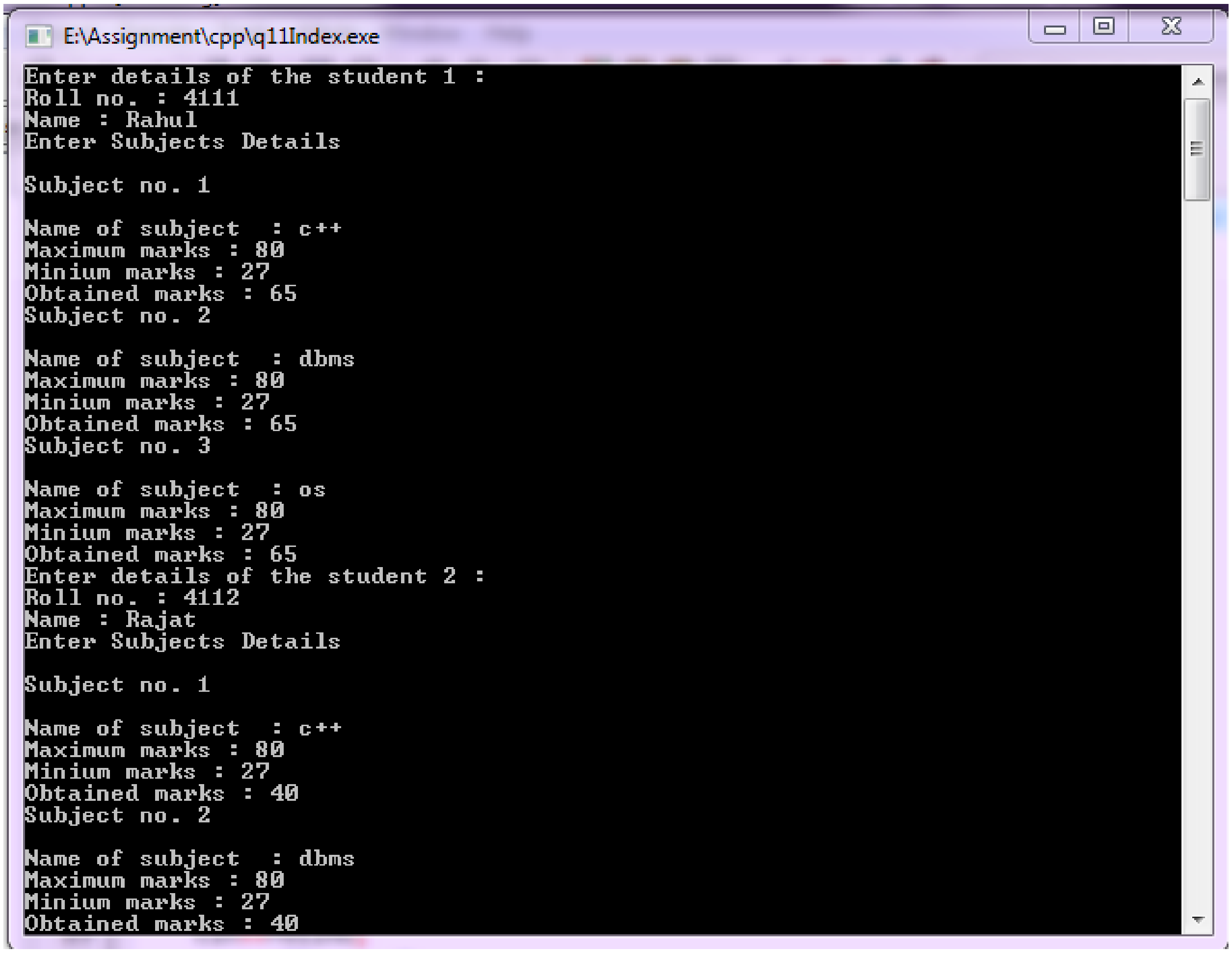
```
#include<iostream>

#include<conio.h>

using namespace std;

class student

{         int rollno;

          char stu_name[20];

          char sub_name[3][20];

          float max_mark[3],min_mark[3],obt_mark[3];

                                                         //data members

             public:

                    void input_data();

                    void disp_spec(int);

                     void disp_result();

                                                  //member function declaration

};
```

```
void student::input_data()                          //member function definition
{       cout<<"Roll no. : ";
        cin>>rollno;
        cout<<"Name : ";
        cin>>stu_name;
        cout<<"Enter Subjects Details "<<endl<<endl;
        int i;
        for(i=0; i<3; i++)
        { cout<<"Subject no. "<<i+1<<endl<<endl;
          cout<<"Name of subject  : ";
          cin>>sub_name[i];
          cout<<"Maximum marks : ";
          cin>>max_mark[i];
          cout<<"Minium marks : ";
          cin>>min_mark[i];
          cout<<"Obtained marks : ";
          cin>>obt_mark[i];
        }
}
void student::disp_spec(int roll)                   //member function definition
{       if(roll==rollno)
        disp_result();      }                       //nesting of member function


void student::disp_result()                         //member function definition
{       cout<<"Result of student "<<stu_name<<"........"<<endl;
```

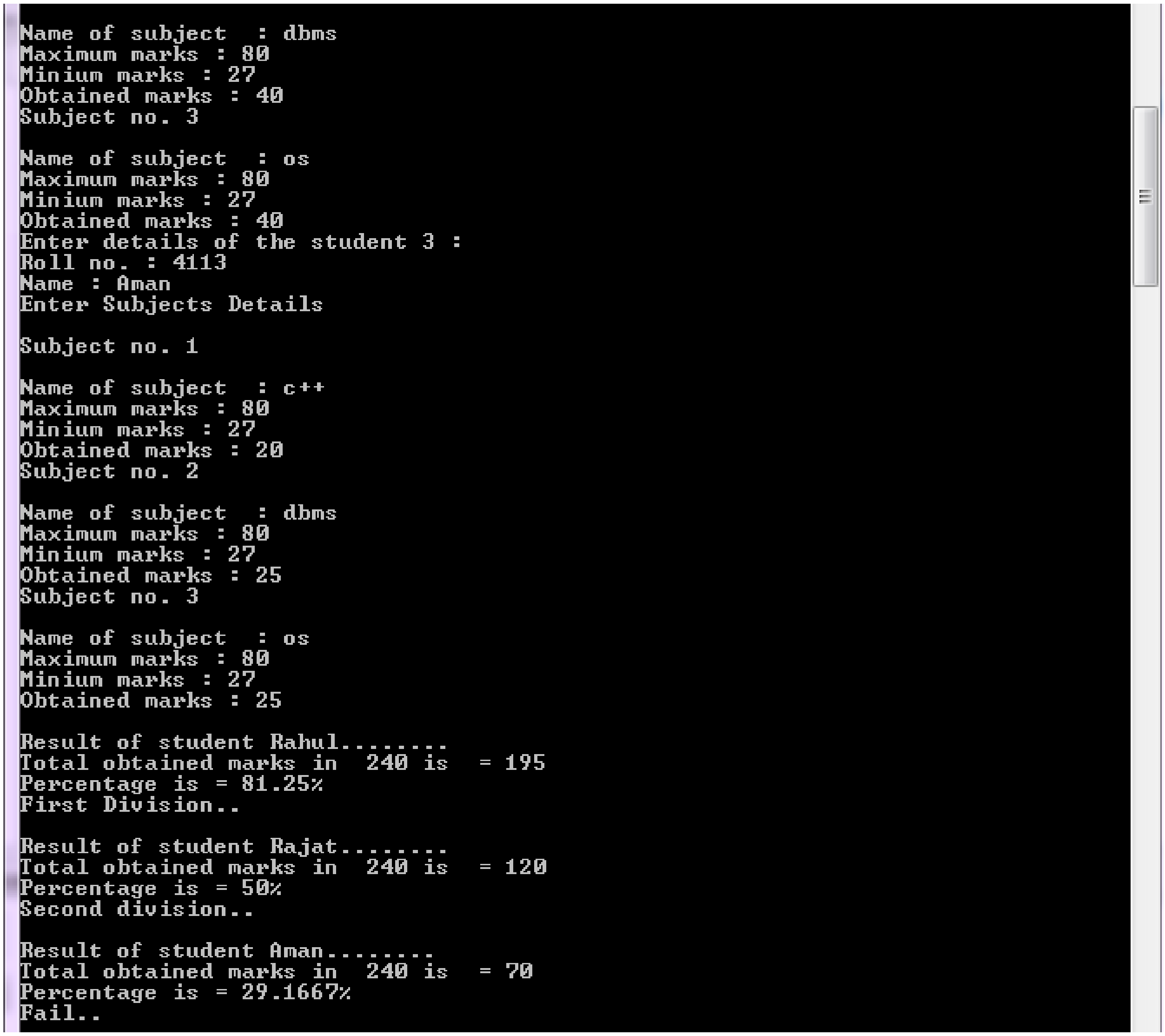
```
        float total_max=0,total_obt=0;
        int i;
        for(i=0; i<3; i++)
        {       total_max = total_max + max_mark[i];
                total_obt = total_obt + obt_mark[i];
        }
        float per;
        per = total_obt * 100 / total_max;
        cout<<"Total obtained marks in  "<<total_max<<" is  = "<<total_obt<<endl;
        cout<<"Percentage is = "<<per<<"%"<<endl;
        if(per>=70)
        cout<<"First Division.."<<endl;
        else if(per<70 && per>=50)
        cout<<"Second division.."<<endl;
        else if(per>=33 && per<50)
        cout<<"Third division..."<<endl;
        else
        cout<<"Fail.."<<endl;
        cout<<endl;         }
int main()
{       student stu[3];                         / /array of object created
        int i;
        for(i=0; i<3; i++)
        {
                cout<<"Enter details of the student "<<i+1<<" : "<<endl;
```

```
                    stu[i].input_data();                    //member function calling
                    }
  cout<<endl;
        for(i=0; i<3; i++)
        stu[i].disp_result();                         //member function calling
        cout<<endl;
  int roll;
        cout<<"Enter Roll no. of student : ";
        cin>>roll;
        for(i=0; i<3; i++)
         stu[i].disp_spec(roll);                    //member function calling
        getch();
        return 0;    }
```

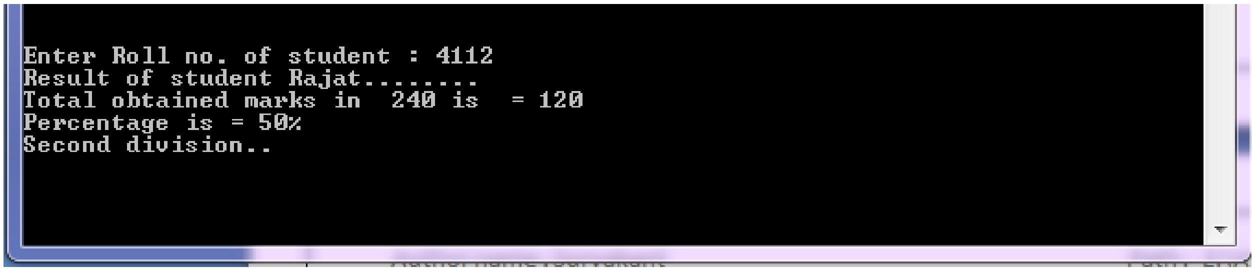
OUTPUT:



```
Enter details of the student 1 :
Roll no. : 4111
Name : Rahul
Enter Subjects Details

Subject no. 1

Name of subject  : c++
Maximum marks : 80
Minium marks : 27
Obtained marks : 65
Subject no. 2

Name of subject  : dbms
Maximum marks : 80
Minium marks : 27
Obtained marks : 65
Subject no. 3

Name of subject  : os
Maximum marks : 80
Minium marks : 27
Obtained marks : 65
Enter details of the student 2 :
Roll no. : 4112
Name : Rajat
Enter Subjects Details

Subject no. 1

Name of subject  : c++
Maximum marks : 80
Minium marks : 27
Obtained marks : 40
Subject no. 2

Name of subject  : dbms
Maximum marks : 80
Minium marks : 27
Obtained marks : 40
```



```
Name of subject  : dbms
Maximum marks : 80
Minium marks : 27
Obtained marks : 40
Subject no. 3

Name of subject  : os
Maximum marks : 80
Minium marks : 27
Obtained marks : 40
Enter details of the student 3 :
Roll no. : 4113
Name : Aman
Enter Subjects Details

Subject no. 1

Name of subject  : c++
Maximum marks : 80
Minium marks : 27
Obtained marks : 20
Subject no. 2

Name of subject  : dbms
Maximum marks : 80
Minium marks : 27
Obtained marks : 25
Subject no. 3

Name of subject  : os
Maximum marks : 80
Minium marks : 27
Obtained marks : 25

Result of student Rahul........
Total obtained marks in  240 is  = 195
Percentage is = 81.25%
First Division..

Result of student Rajat........
Total obtained marks in  240 is  = 120
Percentage is = 50%
Second division..

Result of student Aman........
Total obtained marks in  240 is  = 70
Percentage is = 29.1667%
Fail..
```



```
Enter Roll no. of student : 4112
Result of student Rajat........
Total obtained marks in  240 is  = 120
Percentage is = 50%
Second division..
```

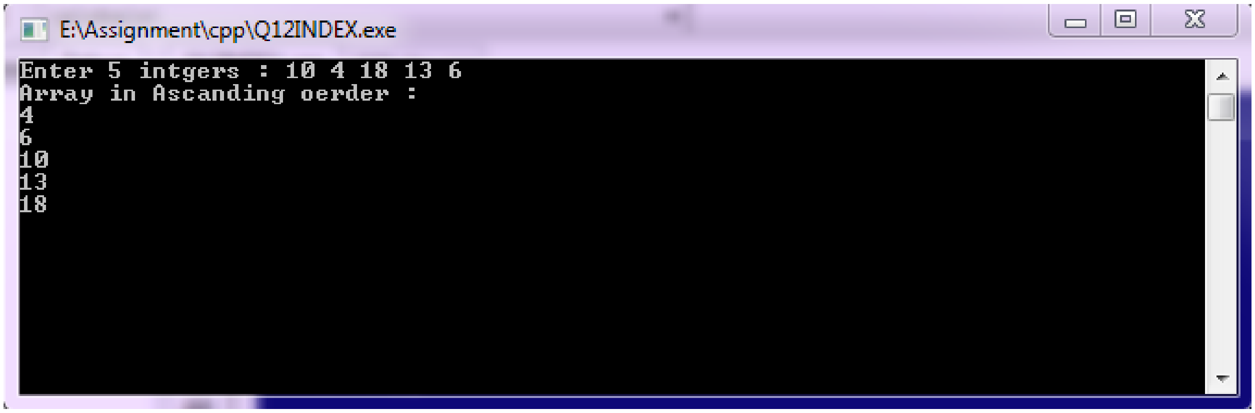
Q12. Create a class named ‘array’ having an array of integers having 5 elements as data member provide following facilities :
Constructor to get number in array elements.

Sort the elements.

PROGRAM :

```
                                    //Program to sort integers
#include<iostream>
#include<conio.h>
using namespace std;
class array
{        int a[5];                                              //data member
        public:
         array(int b[5])                                  //parameterized constructor
          {       int i;
                  for(i=0; i<5; i++)
                  a[i] = b[i];   }
         void sort();
         void swap(int &,int &);                    //member function declaration
};
void array::sort()                                  //member functiion definition
{   int i,j;
    for(i=0; i<5-1; i++)
                  {      for(j=0; j<5-1; j++)
                         {   if(a[j] > a[j+1])
                             swap(a[j],a[j+1]);      }          //swapping the values
                    }
 cout<<"Array in Ascanding oerder : "<<endl;
   for(i=0; i<5; i++)
   cout<<a[i]<<endl;    }
void array::swap(int &a,int &b)                //member functiion definition
{        int temp=a;
        a=b;
        b=temp;       }
int main()
{         int ary[5],i;
        cout<<"Enter 5 intgers : ";
        for(i=0; i<5; i++)
        cin>>ary[i];
        array ary1(ary);                    //creating an object also ary's values passing
        ary1.sort();                        //member function calling
        getch();
        return 0;         }
```

OUTPUT :



```
Enter 5 intgers : 10 4 18 13 6
Array in Ascanding oerder :
4
6
10
13
18
```

Q13. Create a class Static_demo with static member functions for following tasks:-

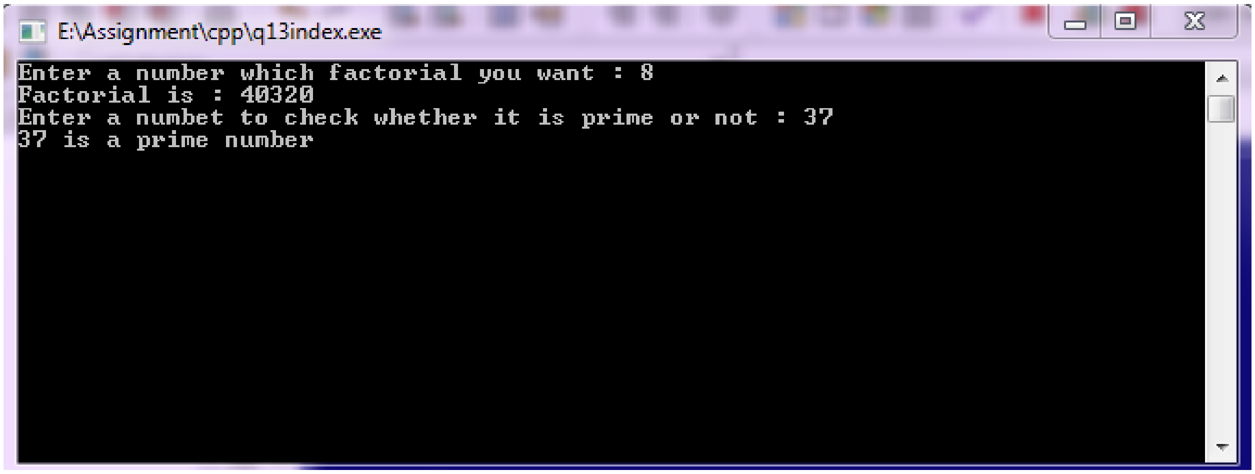
1. To find factorial by recursive member function

2. To check whether a no. is prime or not.

PROGRAM :

```
        //Program to find factorial and check prime number with static member function

#include<iostream>
#include<conio.h>
using namespace std;
class Static_demo
{
        public:
                static double find_facto(double);          //static member function declaration
                static void check_prime(int);              //static member function declaration
};
double Static_demo::find_facto(double n)                   //static member function definition
{
        if( n < 1)
        return 1;
        else
        return n*find_facto(n-1);                 //recursion
}
void Static_demo::check_prime(int a)                       //static member function definition
{
        int i,j;
        int c=0;

        for(i=1; i<=a; i++)
        {
                if(a%i == 0)
                c++;
        }
        if(c<=2)
        cout<<a<<" is a prime number"<<endl;
        else
        cout<<a<<" is not a prime number"<<endl;
}
int main()
{
        double num;
        cout<<"Enter a number which factorial you want : ";
        cin>>num;
        double fact = Static_demo::find_facto(num);
                        //calling static member function of static_demo class
        cout<<"Factorial is : "<<fact<<endl;
        int n;
        cout<<"Enter a numbet to check whether it is prime or not : ";
        cin>>n;
        Static_demo::check_prime(n);
                        //calling static member function
        getch();
        return 0;
}
```

OUTPUT :



```
Enter a number which factorial you want : 8
Factorial is : 40320
Enter a numbet to check whether it is prime or not : 37
37 is a prime number
```

Q14. Write a class complex having data members to store real and imaginary part provide following

Add two complex no using object as an argument.
subtract two complex no using object as an argument.

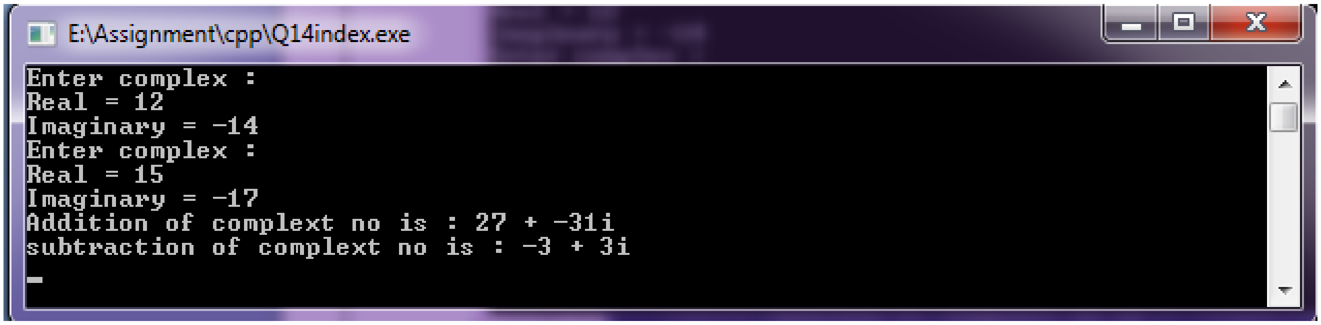
PROGRAM :

//Program to add and subtract two complex number

```
#include<iostream>

#include<conio.h>

using namespace std;

class complex

{    public:

          float real, imag;                          //Data members in public mode

          void get_data();                             //Member function declaration

};

void complex::get_data()                               //member function definition

{        cout<<"Enter complex : "<<endl;

          cout<<"Real = ";

          cin>>real;

          cout<<"Imaginary = ";

          cin>>imag;   }

complex add_complex(complex,complex);            //function prototyping

complex sub_complex(complex,complex);             //function prototyping

complex add_complex(complex c1,complex c2)               //function definition

{   complex temp;

   temp.real = c1.real + c2.real;

   temp.imag = c1.imag + c2.imag;
```

```
  return temp;     }
complex sub_complex(complex c1,complex c2)                          //function definition
{  complex temp;
  temp.real = c1.real - c2.real;
  temp.imag = c1.imag - c2.imag;
  return temp;  }
int main()
{        complex comp1,comp2,add,sub;
          comp1.get_data();
          comp2.get_data();
          add = add_complex(comp1,comp2);   //function calling with passing objects as argument
          cout<<"Addition of complext no is : "<<add.real<<" + "<<add.imag<<"i"<<endl;
          sub = sub_complex(comp1,comp2);   //function calling with passing objects as argument
          cout<<"subtraction of complext no is : "<<sub.real<<" + "<<sub.imag<<"i"<<endl;
          getch();
          return 0;          }
```

OUTPUT :



```
Enter complex :
Real = 12
Imaginary = -14
Enter complex :
Real = 15
Imaginary = -17
Addition of complext no is : 27 + -31i
subtraction of complext no is : -3 + 3i
_
```

Q15. Write swapping program to demonstrate call by value , call by address and call by reference in a single program ?

PROGRAM :

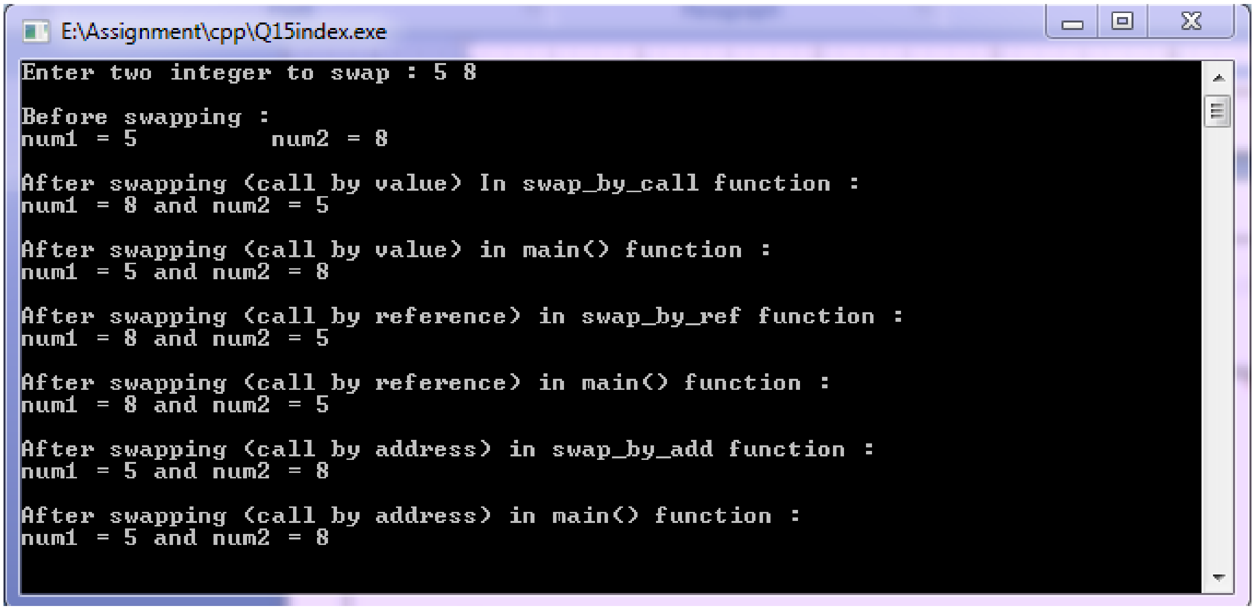
//Swapping program to demonstrate call by value, call by reference and call by address

```
#include<iostream>

#include<conio.h>

using namespace std;

void swap_by_call(int,int);

void swap_by_ref(int &,int &);

void swap_by_add(int *,int *);

                                                    // function prototyping

void swap_by_call(int a,int b)                        //function declaration

{

        int temp = a;

        a = b;

        b = temp;

  cout<<"After swapping (call by value) In swap_by_call function : "<<endl;

  cout<<"num1 = "<<a<<" and num2 = "<<b<<endl<<endl;

}

void swap_by_ref(int &a,int &b)                     //function declaration

{

        int temp = a;

        a = b;

        b = temp;

        cout<<"After swapping (call by reference) in swap_by_ref function : "<<endl;

   cout<<"num1 = "<<a<<" and num2 = "<<b<<endl<<endl;
```

```
}
void swap_by_add(int *a,int *b)                         //function declaration
{
        int temp = *a;
        *a = *b;
        *b = temp;
   cout<<"After swapping (call by address) in swap_by_add function : "<<endl;
  cout<<"num1 = "<<*a<<" and num2 = "<<*b<<endl<<endl;
}
int main()
{
        int x,y;
        cout<<"Enter two integer to swap : ";
        cin>>x>>y;
        cout<<endl;
        cout<<"Before swapping : "<<endl;
        cout<<"num1 = "<<x<<"\t num2 = "<<y<<endl<<endl;
        swap_by_call(x,y);                                //function calling
 cout<<"After swapping (call by value) in main() function : "<<endl;
 cout<<"num1 = "<<x<<" and num2 = "<<y<<endl<<endl;
        swap_by_ref(x,y);                                 //function calling
 cout<<"After swapping (call by reference) in main() function : "<<endl;
 cout<<"num1 = "<<x<<" and num2 = "<<y<<endl<<endl;
        swap_by_add(&x,&y);                               //function calling
 cout<<"After swapping (call by address) in main() function : "<<endl;
```

```
  cout<<"num1 = "<<x<<" and num2 = "<<y<<endl<<endl;

   getch();

return 0;

}
```

OUTPUT :



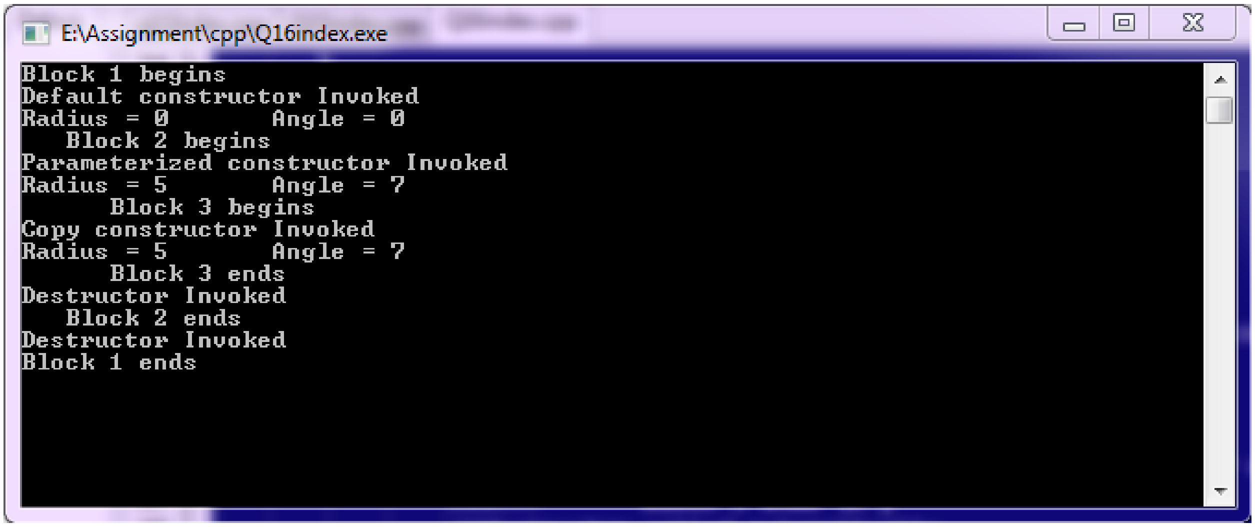
```
Enter two integer to swap : 5 8

Before swapping :
num1 = 5          num2 = 8

After swapping (call by value) In swap_by_call function :
num1 = 8 and num2 = 5

After swapping (call by value) in main() function :
num1 = 5 and num2 = 8

After swapping (call by reference) in swap_by_ref function :
num1 = 8 and num2 = 5

After swapping (call by reference) in main() function :
num1 = 8 and num2 = 5

After swapping (call by address) in swap_by_add function :
num1 = 5 and num2 = 8

After swapping (call by address) in main() function :
num1 = 5 and num2 = 8
```

Q16. Write a program for to create class polar data member radius and angle define constructor of all three types and create destructor and test function in main.

PROGRAM :

```
                    //Program to demonstrate constructors and destrutor

#include<iostream>

#include<conio.h>

using namespace std;

class polar                                                    //class definition

{

 float radius;

 float angle;

   public:

        polar()                                         //default construtor definition

        {     radius = 0;

               angle  = 0;

               cout<<"Default constructor Invoked \n";

        cout<<"Radius = "<<radius<<"\t Angle = "<<angle<<endl;

        }

        polar(float r,float a)                      //parameterized constructor definition

        {     radius = r;

               angle = a;

               cout<<"Parameterized constructor Invoked \n";
        cout<<"Radius = "<<radius<<"\t Angle = "<<angle<<endl;
     }
     polar(polar &p)                              //copy consttructor definition
     {  radius = p.radius;
        angle = p.angle;
        cout<<"Copy constructor Invoked  "<<endl;
        cout<<"Radius = "<<radius<<"\t Angle = "<<angle<<endl;
     }
     ~polar()                                     //Destructor definition
     {
        cout<<"Destructor Invoked"<<endl;
     }
};
int main()
{    cout<<"Block 1 begins \n";
    polar p1;                  //object created and default constructor called
    {
        cout<<"   Block 2 begins \n";
        polar p2(5,7);
              //object created and parameterized constructor called
         {
            cout<<"     Block 3 begins \n";
             polar p3 = p2;         //object created annd copy constructor called
            cout<<"     Block 3 ends \n";
                   //destructor called automatically
        }
        cout<<"   Block 2 ends \n";
                                      //destructor called automatically
    }
    cout<<"Block 1 ends \n";
                                      //destructor called automatically
    getch();
    return 0;
}
```

OUTPUT :



```
Block 1 begins
Default constructor Invoked
Radius = 0        Angle = 0
   Block 2 begins
Parameterized constructor Invoked
Radius = 5        Angle = 7
      Block 3 begins
Copy constructor Invoked
Radius = 5        Angle = 7
      Block 3 ends
Destructor Invoked
   Block 2 ends
Destructor Invoked
Block 1 ends
```

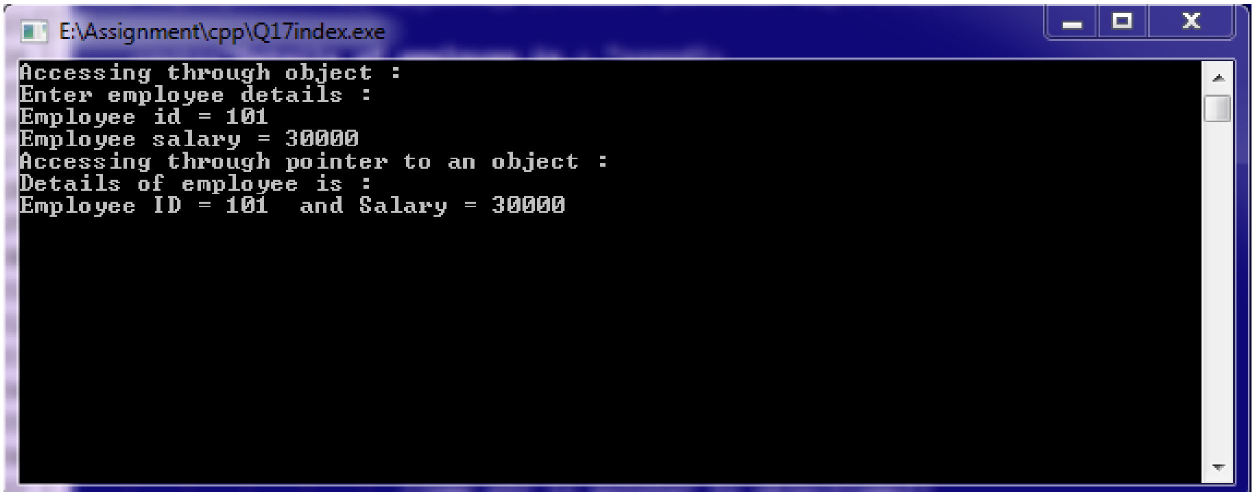
Q17. WAP to create a class employee having data member employed id,salary.proide member function for data input,output,use pointer to an object information of employee and test the program in function main ?

PROGRAM :

//program to access the member functions of a class by pointer to an object

```
#include<iostream>
#include<conio.h>
using namespace std;
class employee                                                    //class definition
{
        int emp_id;
        float salary;
                                                                  //data members
        public:              //public area
          void get_emp_data();                          //member function declaration
          void disp_emp_data();                         //member function declaration
};

void employee::get_emp_data()                          //member function definition
{
        cout<<"Enter employee details : "<<endl;
        cout<<"Employee id = ";
        cin>>emp_id;
        cout<<"Employee salary = ";
        cin>>salary;
}

void employee::disp_emp_data()                          //member function definition
{
        cout<<"Details of employee is : "<<endl;
        cout<<"Employee ID = "<<emp_id<<"  and Salary = "<<salary<<endl;
}

int main()
{
        employee emp1;
                          //object emp1 created of employee class
        cout<<"Accessing through object : "<<endl;
        emp1.get_emp_data();
                             //data entered with the help of object emp1

        employee *emp_ptr;
                         //creating pointer emp_ptr of type employee
        emp_ptr = &emp1;                    //emp_ptr holds address of object emp1
                                            //emp_ptr is pointer to object(emp1)
        cout<<"Accessing through pointer to an object : "<<endl;
        emp_ptr->disp_emp_data();
                    //Displaying data stored in employee class via emp_ptr (pointer to an object)
        getch();
        return 0;
}
```

OUTPUT :



```
Accessing through object :
Enter employee details :
Employee id = 101
Employee salary = 30000
Accessing through pointer to an object :
Details of employee is :
Employee ID = 101  and Salary = 30000
```

Q18. Write program-using class and to store data about books(book id,Title,Author,Price,Edition)

provide following facilities :

Addition of new books.
Searching for availability of books if provide author.

PROGRAM :

//Program to add new book and search the book

```
#include<iostream>

#include<conio.h>

#include<string.h>

using namespace std;


class books

{

  int book_id;

        char title[20];

        char author[20];

        float price;

        char edition[20];                              //data members

        public:

                void add_book();

                int search_book(char);

                void display_books();            //member function declaration

};
```

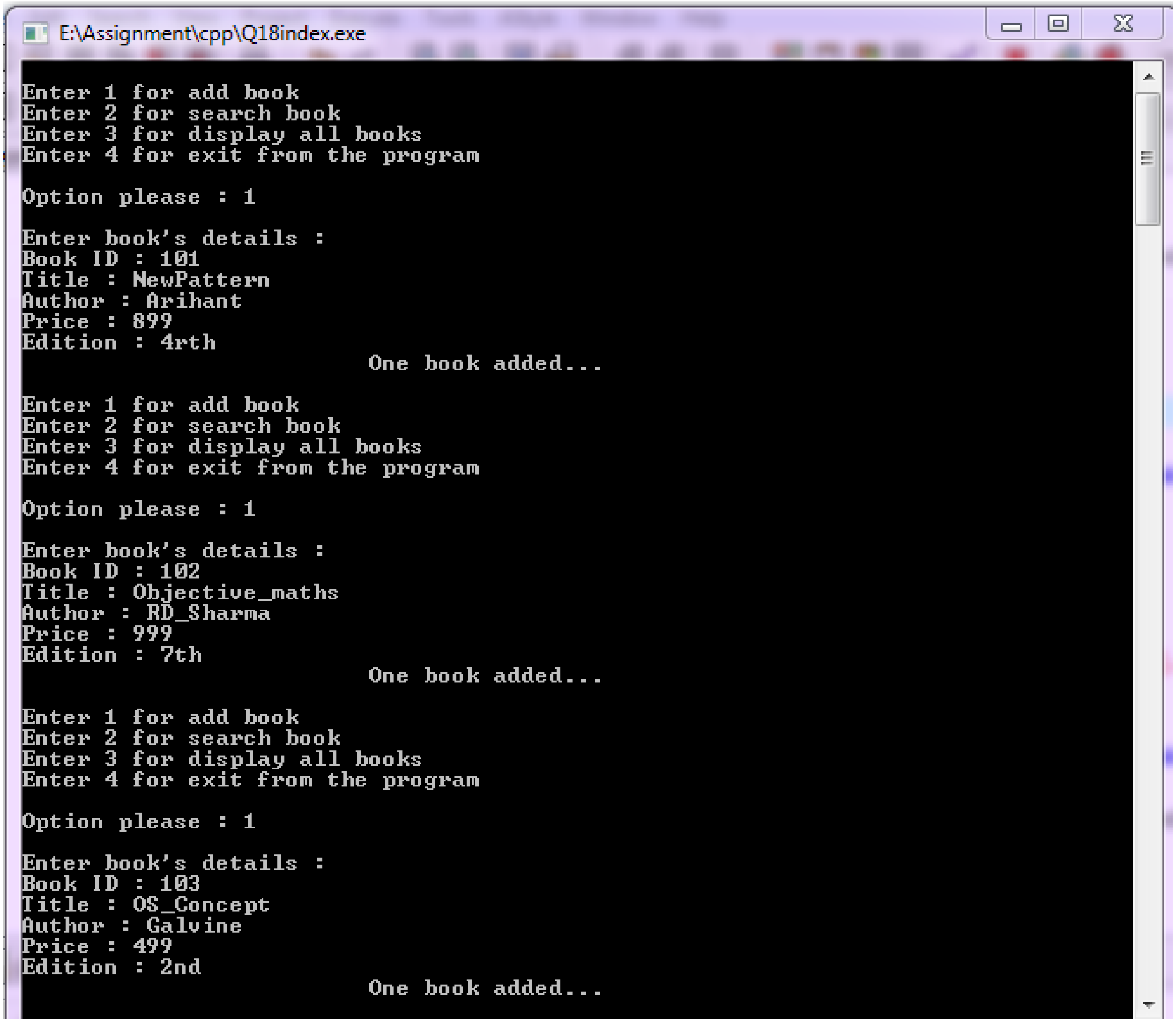
```
void books::add_book()                                  //member fun. definition
{       cout<<"Enter book's details : "<<endl;
          cout<<"Book ID : ";
          cin>>book_id;
          cout<<"Title : ";
          cin>>title;
          cout<<"Author : ";
          cin>>author;
          cout<<"Price : ";
          cin>>price;
          cout<<"Edition : ";
          cin>>edition;
}
int books::search_book(char tmp_author[20])             //member fun. definition
{
   if(strcmp(tmp_author,author)
   {  display_books();
          return 1;  }
   else
  return 0;     }
void books::display_books()                        //member fun. definition
 {        cout<<"Books deatails....."<<endl<<endl;
          cout<<"Book ID : "<<book_id;
          cout<<"\t Title : "<<title<<endl;
          cout<<"Author : "<<author;
```

```
            cout<<"\t Price : "<<price<<endl;

            cout<<"Edition : "<<edition<<endl;   }

    void loop(char c)

    {  int j;

              char ch;

              ch = c;

              for(j=0; j<80; j++)

              cout<<ch;  }

    int main()

    {          books *ptr,book[20];

            ptr = book;

            int inc;

            int total_books=0;

            int k;

            do{

                    int op;

                    cout<<endl;

                    cout<<"Enter 1 for add book"<<endl;

                    cout<<"Enter 2 for search book"<<endl;

                    cout<<"Enter 3 for display all books"<<endl;

                    cout<<"Enter 4 for exit from the program "<<endl;

                    cout<<endl<<"Option please : ";

                    cin>>op;

                    cout<<endl;

                     inc=0;
            switch(op)
        {
          case 1:                                    //add book
                    {       book[total_books].add_book();
                       cout<<"\t\t\t One book added..."<<endl;
                      inc++;
                    }
              break;
                    case 2:                                        //search book
              {
                 string tmp_auth;
                  cout<<"Who is the author of the book? please enter : ";
                  cin>>tmp_auth;
                  int i;
                  int found_count=0;
                  for(i=0; i<total_books; i++)
                  found_count = found_count + book[i].search_book(tmp_auth);
                            if(found_count > 0)
                            cout<<"\t\t\t "<<found_count<<" book(s) found..."<<endl;
                            else
                            cout<<"\t\t\t No book found..."<<endl;
              }
             break;
              case 3:                                        //display all book
                    {      for(k=0; k<total_books; k++)
```

```
                         {   loop('-');
                               cout<<"Book "<<k+1<<endl;
                               book[k].display_books();
                               loop('-');
                               cout<<endl;
                          }
               }
                  break;
                  case 4:
                  exit(1); //exit statement
                  break;
                   default:
                   cout<<"Choose right option "<<endl;
  }                                         //switch case ends
 total_books = total_books + inc;
 }while(1);                            //do...while ends
 getch();
  return 0;
}
```

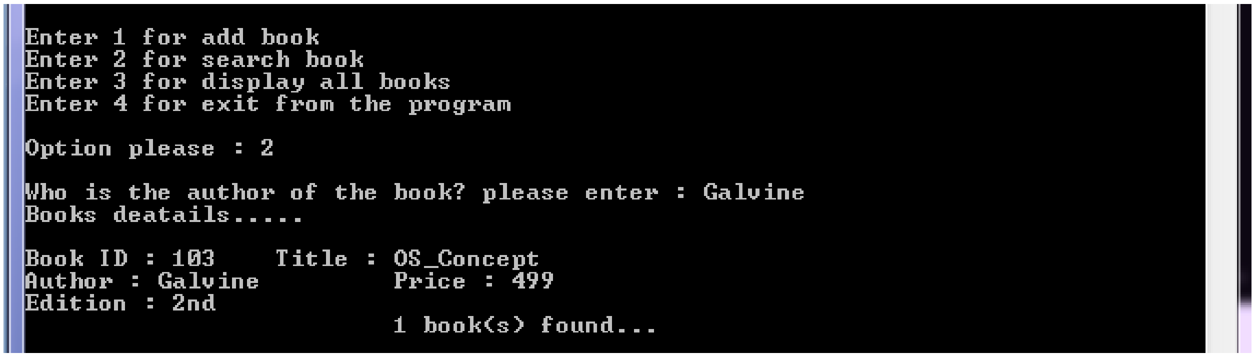
OUTPUT :

Adding books :



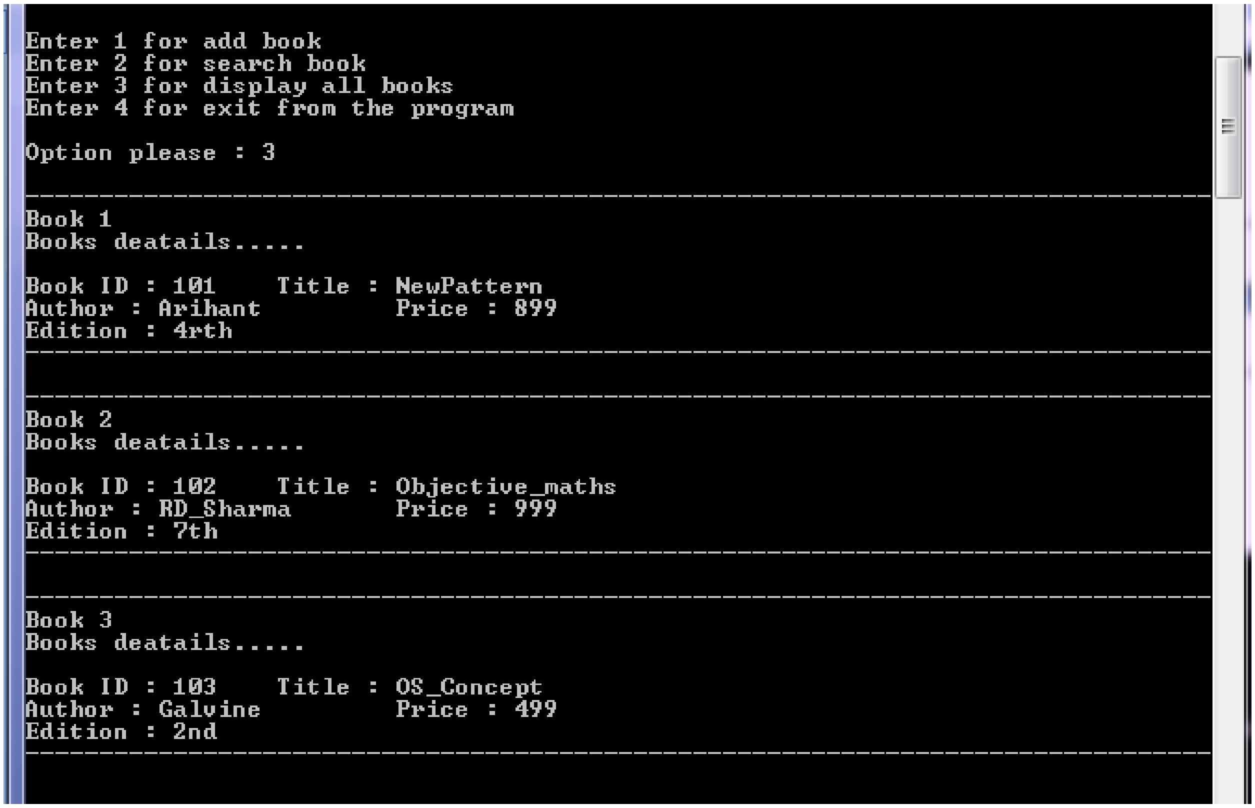
```
Enter 1 for add book
Enter 2 for search book
Enter 3 for display all books
Enter 4 for exit from the program

Option please : 1

Enter book's details :
Book ID : 101
Title : NewPattern
Author : Arihant
Price : 899
Edition : 4rth
                         One book added...

Enter 1 for add book
Enter 2 for search book
Enter 3 for display all books
Enter 4 for exit from the program

Option please : 1

Enter book's details :
Book ID : 102
Title : Objective_maths
Author : RD_Sharma
Price : 999
Edition : 7th
                         One book added...

Enter 1 for add book
Enter 2 for search book
Enter 3 for display all books
Enter 4 for exit from the program

Option please : 1

Enter book's details :
Book ID : 103
Title : OS_Concept
Author : Galvine
Price : 499
Edition : 2nd
                         One book added...
```

Searching book by Author name :



```
Enter 1 for add book
Enter 2 for search book
Enter 3 for display all books
Enter 4 for exit from the program

Option please : 2

Who is the author of the book? please enter : Galvine
Books deatails.....

Book ID : 103    Title : OS_Concept
Author : Galvine         Price : 499
Edition : 2nd
                         1 book(s) found...
```

Displaying all books :



```
Enter 1 for add book
Enter 2 for search book
Enter 3 for display all books
Enter 4 for exit from the program

Option please : 3

--------------------------------------------------------------------------------
Book 1
Books deatails.....

Book ID : 101    Title : NewPattern
Author : Arihant         Price : 899
Edition : 4rth
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Book 2
Books deatails.....

Book ID : 102    Title : Objective_maths
Author : RD_Sharma       Price : 999
Edition : 7th
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Book 3
Books deatails.....

Book ID : 103    Title : OS_Concept
Author : Galvine         Price : 499
Edition : 2nd
--------------------------------------------------------------------------------
```

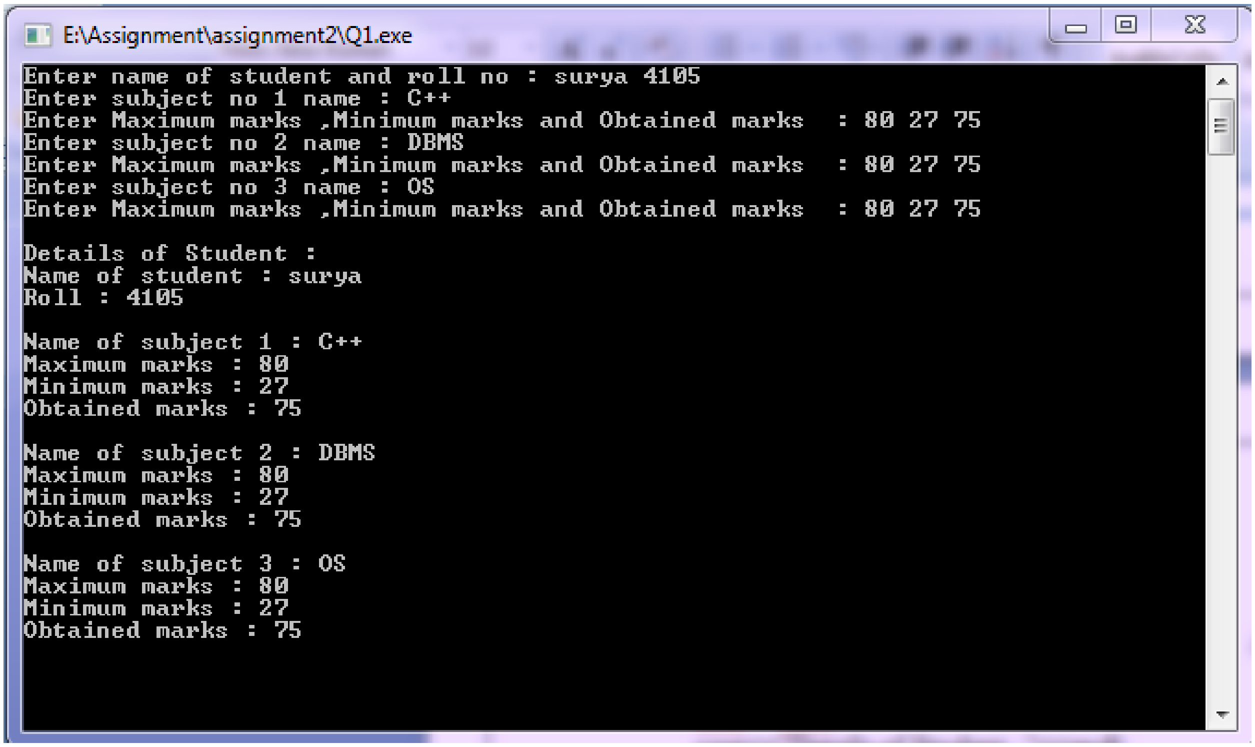
Q19. **Define structure student. Structure student has data members for storing name, rollno, name of three subjects and marks. Write member function to store and print data.**

PROGRAM :

```
                    //program to store the data  of a structure student and print them

#include<iostream>
#include<conio.h>
using namespace std;
struct student
{
        char name[20];
        int rollno;
        char sub_name[3][20];
        float max_marks[3],min_marks[3],obt_marks[3];
                                                        //Data members of structure
        void getdata()
        {
                 cout<<"Enter name of student and roll no : ";
                 cin>>name>>rollno;
                 int i;
                 for(i=0; i<3; i++)
                 {
                        cout<<"Enter subject no "<<i+1<<" name : ";
                        cin>>sub_name[i];
                        cout<<"Enter Maximum marks ,Minimum marks and Obtained marks  : ";
                        cin>>max_marks[i]>>min_marks[i]>>obt_marks[i];
                 }
     }
                void disp_data()
                {
                        cout<<"Name of student : "<<name<<endl;
                        cout<<"Roll : "<<rollno<<endl;
                        int i;
                        for(i=0; i<3; i++)
                        {   cout<<endl;
                                cout<<"Name of subject "<<i+1<<" : "<<sub_name[i]<<endl;
                                cout<<"Maximum marks : "<<max_marks[i]<<endl;
                                cout<<"Minimum marks : "<<min_marks[i]<<endl;
                                cout<<"Obtained marks : "<<obt_marks[i]<<endl;
                        }
                }
                                        //member function definitions of structure
};
```

```
int main()
{
        struct student s1;                          //structure variable created
        s1.getdata();
        cout<<endl;
        cout<<"Details of Student : "<<endl;
        s1.disp_data();
                                            //accessing the member functions of structure
        getch();

        return 0;
}
```

OUTPUT :



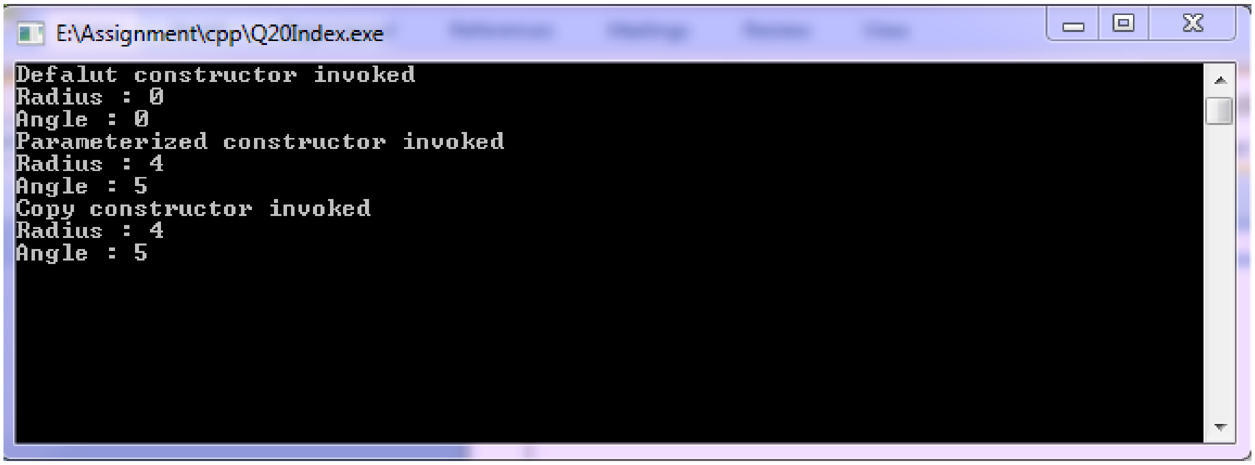
```
Enter name of student and roll no : surya 4105
Enter subject no 1 name : C++
Enter Maximum marks ,Minimum marks and Obtained marks  : 80 27 75
Enter subject no 2 name : DBMS
Enter Maximum marks ,Minimum marks and Obtained marks  : 80 27 75
Enter subject no 3 name : OS
Enter Maximum marks ,Minimum marks and Obtained marks  : 80 27 75

Details of Student :
Name of student : surya
Roll : 4105

Name of subject 1 : C++
Maximum marks : 80
Minimum marks : 27
Obtained marks : 75

Name of subject 2 : DBMS
Maximum marks : 80
Minimum marks : 27
Obtained marks : 75

Name of subject 3 : OS
Maximum marks : 80
Minimum marks : 27
Obtained marks : 75
```

Q20. Write program to create a class Polar which has data member radius and angle, define overloaded constructor to initialize object and copy constructor to initialize one object by another existing object keep name of parameter of parameterized constructor same as data members. Test function of the program in main function.

PROGRAM :

```
                    //Programm to demonstrate constructor overloading

#include<iostream>

#include<conio.h>

using namespace std;

class Polar

{

        float radius;

        float angle;

  public:

                Polar()                                 //default constructor

                {

                        radius = 0;

                        angle = 0;

                        cout<<"Defalut constructor invoked "<<endl;

                }

                Polar(float radius,float angle)                 //parameterized constructor

                {                               //formal argument is same as data member

                        this->radius = radius;

                        this->angle = angle;        //this poiter is used to avoid conflict

                       cout<<"Parameterized constructor invoked "<<endl;

                }
                Polar(Polar &p)                                   //copy constructor
                {
                        radius = p.radius;
                        angle = p.angle;
                         cout<<"Copy constructor invoked "<<endl;
                }
                                            //contructor overloaded
                void display()
                {
                        cout<<"Radius : "<<radius<<endl;
                        cout<<"Angle : "<<angle<<endl;
                }
};
int main()
{
        Polar p1;
           //default constructor invoked
        p1.display();
        Polar p2(4,5);
          //Parameterized constructor invoked
        p2.display();
        Polar p3 = p2;
          //copy constructor invoked
        p3.display();
        getch();
        return 0;

}
```

OUTPUT :



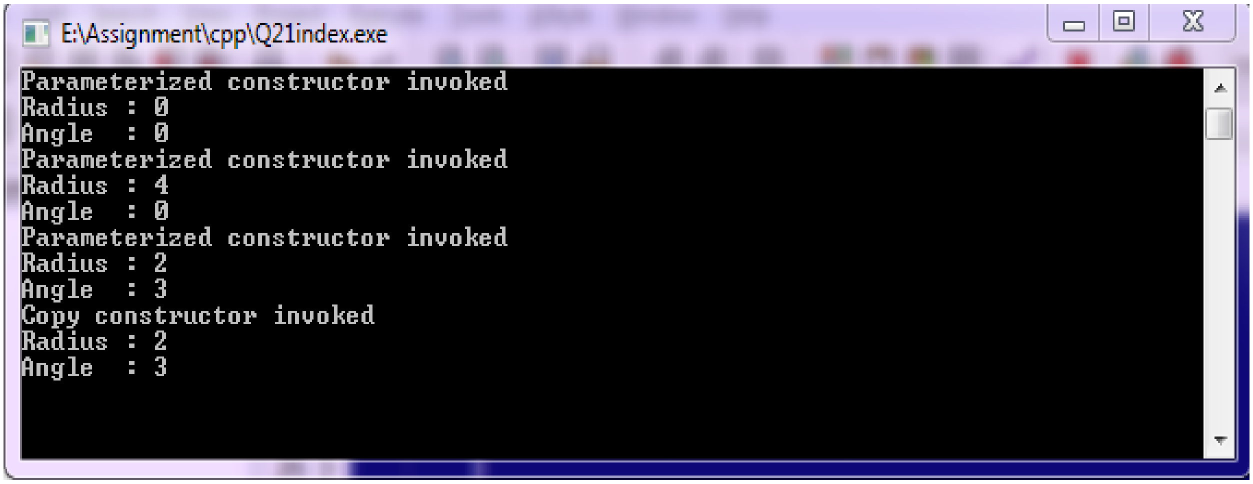
```
Defalut constructor invoked
Radius : 0
Angle : 0
Parameterized constructor invoked
Radius : 4
Angle : 5
Copy constructor invoked
Radius : 4
Angle : 5
```

Q21. Write program to create a class Polar which has data member radius and angle, use constructor with default arguments to avoid constructor overloading and copy constructor to initialize one object by another existing object keep name of parameter of parameterized constructor same as data members. Test functioning of the program in main function.

PROGRAM :

```
                    //Programm to demonstrate constructor with default arguments

#include<iostream>

#include<conio.h>

using namespace std;

class Polar

{         float radius;

          float angle;

   public:

          Polar(float radius=0,float angle=0) //parameterized constructor with defalut arguments

                    {                    //formal argument is same as data member

                              this->radius = radius;

                              this->angle = angle;  //this poiter is used to avoid conflict

               cout<<"Parameterized constructor invoked "<<endl;

                    }

                    Polar(Polar &p) //copy constructor

                    {         radius = p.radius;

                              angle = p.angle;

                               cout<<"Copy constructor invoked "<<endl;

                    }

                    void display()
                    {       cout<<"Radius : "<<radius<<endl;
                              cout<<"Angle  : "<<angle<<endl;   }
};
int main()
{       Polar p1;           //default argument parameterized constructor invoked
          p1.display();
          Polar p2(4);        //default argument Parameterized constructor invoked
          p2.display();
          Polar p3(2,3);          //Parameterized constructor invoked
          p3.display();
          Polar p4 = p3;          //copy constructor invoked
          p3.display();
          getch();
          return 0;
}
```

OUTPUT :



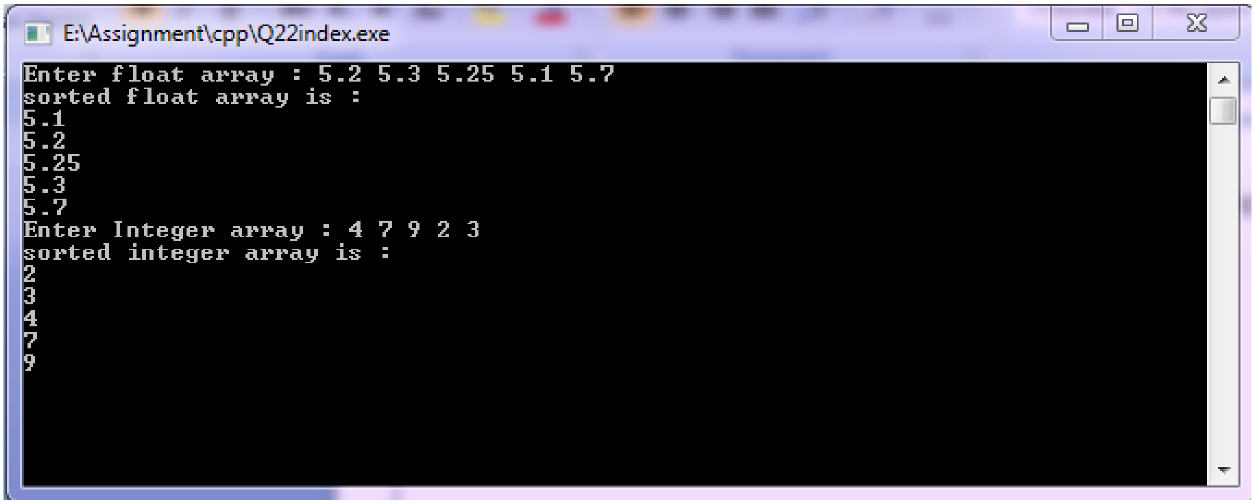
```
Parameterized constructor invoked
Radius : 0
Angle  : 0
Parameterized constructor invoked
Radius : 4
Angle  : 0
Parameterized constructor invoked
Radius : 2
Angle  : 3
Copy constructor invoked
Radius : 2
Angle  : 3
```

Q22. Write a class ArraySort that uses static overloaded function to sort an array of floats, an array of integers.

PROGRAM :

```
                //Program to sort array of float and int using static overloaded function

#include<iostream>
#include<conio.h>
#include<string.h>
using namespace std;

class ArraySort
{
  public:
static void sort_array(float f[5]);
static void sort_array(int f[5]);                    //static member function overloading
static void swap(float &a,float &b);
static void swap(int &a,int &b);
};
    void ArraySort :: sort_array(float f[5])                    //static member function definition
   {
            int i,j;
             for(i=0; i<5-1; i++)
             {
                        for(j=0; j<5-1; j++)
                {
                if(f[j] > f[j+1])
                swap(f[j],f[j+1]);                              //nesting static member function
                }

              }
             cout<<"sorted float array is : "<<endl;
             for(i=0; i<5; i++)
             cout<<f[i]<<endl;
    }
  Void ArraySort :: sort_array(int f[5])                        //static member function overloading
   {
            int i,j;
             for(i=0; i<5-1; i++)
             {
                        for(j=0; j<5-1; j++)
                {

                        if(f[j] > f[j+1])
                        swap(f[j],f[j+1]);                      //nesting static member function
              }

   }


            cout<<"sorted integer array is : "<<endl;
            for(i=0; i<5; i++)
           cout<<f[i]<<endl;
 }
 void ArraySort :: swap(float &a,float &b)
   {
            float temp = a;
            a = b;
            b = temp;
   }
 Void ArraySort :: swap(int &a,int &b)
   {
            int temp = a;
            a = b;
            b = temp;
   }
                                                              //member functions definition
int main()
{
           ArraySort a1;
           float fary[5];
           cout<<"Enter float array : ";
           int i;
           for(i=0; i<5; i++)
           cin>>fary[i];

           a1.sort_array(fary);
                                               // static member function calling and passing float array
                                             //sort_array(float) invoked
           int ary[5];
           cout<<"Enter Integer array : ";
           for(i=0; i<5; i++)
           cin>>ary[i];
           a1.sort_array(ary);
                                          // static member function calling and passing int array
                                          //sort_array(int) invoked
           getch();
           return 0;
}
```

OUTPUT :



```
Enter float array : 5.2 5.3 5.25 5.1 5.7
sorted float array is :
5.1
5.2
5.25
5.3
5.7
Enter Integer array : 4 7 9 2 3
sorted integer array is :
2
3
4
7
9
```

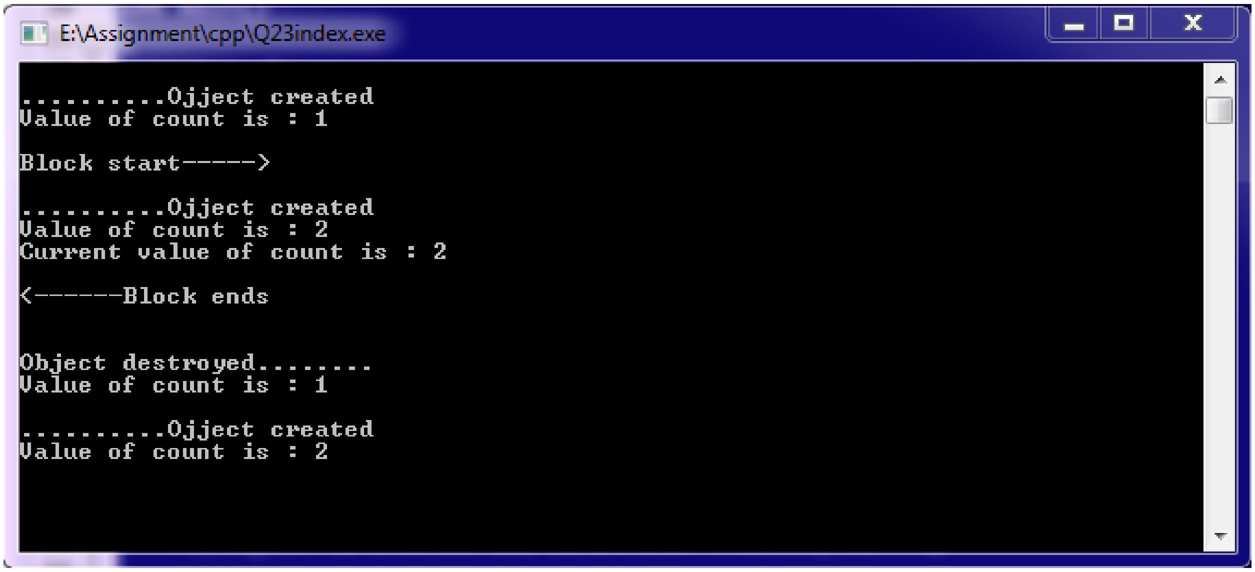
Q23. Create a class Counter having a static data member, which keeps track of no. of objects created of type Counter. One static member function must be created to increase value of static data member as the object is created. One static member function must be created to decrease value of static data member as the object is destroyed. One static member function must be created to display the current value of static data member. Use main function to test the class Counter.

PROGRAM :

//Program to make a class counter having static member functions to keep track count of object (current,when created,when destroyed)

```
#include<iostream>
#include<conio.h>
using namespace std;
class counter
{
        static int count_obj;
        public:
                 counter()                                    // default constructor
                {
                        cout<<endl<<"..........Ojject created"<<endl;
                        inc_count();
                                                      //nested static member function calling
                }
                ~counter()                                    //destructor
                {
                        cout<<endl<<"Object destroyed........"<<endl;
                        dec_count();
                                                       //nested static member function calling
                }
        static void inc_count()
        {
                count_obj++;
                cout<<"Value of count is : "<<count_obj<<endl;
        }
        static void dec_count()
       {
        count_obj--;
        cout<<"Value of count is : "<<count_obj<<endl;
        }
        static void cur_count()
        {
         cout<<"Current value of count is : "<<count_obj<<endl;
        }
                                                 //static member functions definition
};
int counter::count_obj;
int main()
{      counter c1;
                                   //object c1 created and constructor invoked (count = 1)
 {
  cout<<endl<<"Block start-----> "<<endl;
  counter c2;
                                    //object c2 created and constructor invoked (count = 2)
  counter::cur_count();                              //current value (count =2)
                                             //calling static member function
 cout<<endl<<"<------Block ends"<<endl<<endl;
 }
                                  //object c2 destroyed and destrutor invoked (count = 1)


 counter c3;
                             //object c3 created and construtor invoked(count=2)

                         //object c3 destroyed and destrutor invoked (count = 1)
                 //object c1 destroyed and destrutor invoked (count = 0)
   getch();
return 0;
}
```

OUTPUT :



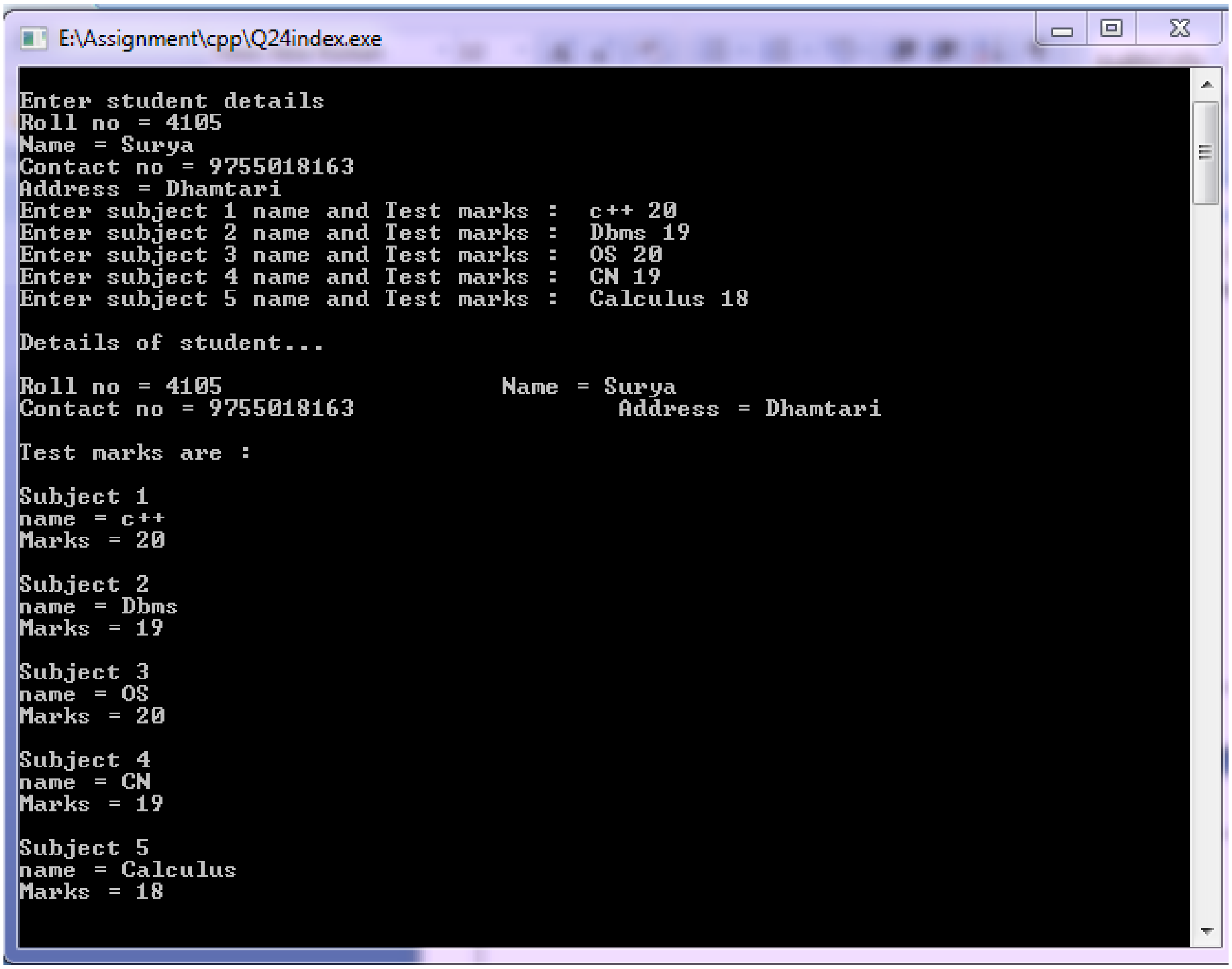
```
..........Ojject created
Value of count is : 1

Block start----->

..........Ojject created
Value of count is : 2
Current value of count is : 2

<------Block ends


Object destroyed.........
Value of count is : 1

..........Ojject created
Value of count is : 2
```

Q24. Create a class student. The student class has data members such as roll number, name of student,contact number and address .create the derived class test which contains data members reperesenting name of subject, and test marks of 5 subjects. Display all the information of student.

PROGRAM :

```
                    //program to store data about student and test using class

#include<iostream>
#include<conio.h>
using namespace std;

class student
{
int rollno;
char name[20];
char contact_no[20];
char addr[20];
                                                    //Data members
public:
        void get_student_data();
        void display_student_data();
                                                    //member function declaration
};
class test:public student                     //derived class test from base calss student
{
  char sub_name[5][20];
  float marks[5];
                                                        //Data members
    public:
            void get_test_data();
            void display_test_data();
                                                    //member function declaration

};
void student::get_student_data()              //student class member function definition
{   cout<<endl;
        cout<<"Enter student details"<<endl;
        cout<<"Roll no = ";
        cin>>rollno;
        cout<<"Name = ";
        cin>>name;
        cout<<"Contact no = ";
        cin>>contact_no;
        cout<<"Address = ";
        cin>>addr;
}
void student::display_student_data()                //student class member function definition
{
        cout<<endl<<"Details of student... "<<endl<<endl;
        cout<<"Roll no = "<<rollno<<"\t\t\t Name = "<<name<<endl;
        cout<<"Contact no = "<<contact_no<<"\t\t\t Address = "<<addr<<endl;
}
void test::get_test_data()              //test class member function definition
{
        int i;
        for(i=0; i<5; i++)
        {
                cout<<"Enter subject "<<i+1<<" name and Test marks :  ";
                cin>>sub_name[i]>>marks[i];
        }
}
void test::display_test_data()          //test class member function definition
{
        int i;
        cout<<endl<<"Test marks are :"<<endl<<endl;
        for(i=0; i<5; i++)
        {

                cout<<"Subject "<<i+1<<endl;
                cout<<"name = "<<sub_name[i]<<endl;
                cout<<"Marks = "<<marks[i]<<endl;
                cout<<endl;
        }
}
int main()
{
        test t1;                        // Crteating object of derived class
        t1.get_student_data();          //calling base class member function
        t1.get_test_data();              //own member function calling

        t1.display_student_data();              //calling base class member function
        t1.display_test_data();                  //own member function calling
        getch();
        return 0;
}
```

OUTPUT :



```
Enter student details
Roll no = 4105
Name = Surya
Contact no = 9755018163
Address = Dhamtari
Enter subject 1 name and Test marks :  c++ 20
Enter subject 2 name and Test marks :  Dbms 19
Enter subject 3 name and Test marks :  OS 20
Enter subject 4 name and Test marks :  CN 19
Enter subject 5 name and Test marks :  Calculus 18

Details of student...

Roll no = 4105                   Name = Surya
Contact no = 9755018163                  Address = Dhamtari

Test marks are :

Subject 1
name = c++
Marks = 20

Subject 2
name = Dbms
Marks = 19

Subject 3
name = OS
Marks = 20

Subject 4
name = CN
Marks = 19

Subject 5
name = Calculus
Marks = 18
```

Q25. Write a program in c++ for multiple inheritance using book as derived class having different base classes Journals,Magzines,Newpaper.

PROGRAM :

```
                    //Program to demonstrate multiple inheritence

#include<iostream>

#include<conio.h>

using namespace std;

class Journals                                      //base class 1 definition

{       char journal_name[20];

        float price;

        public:

           void get_data()

         {   cout<<endl<<"Enter journal name : ";

           cin>>journal_name;

           cout<<"Enter price : ";

           cin>>price;

         }

        void disp_data()

        {     cout<<endl<<"journal name : "<<journal_name<<endl;

                cout<<"price : "<<price<<endl;

        }

                            //member function definitions

};

class Magzines                  //base class 2 definition

{
```

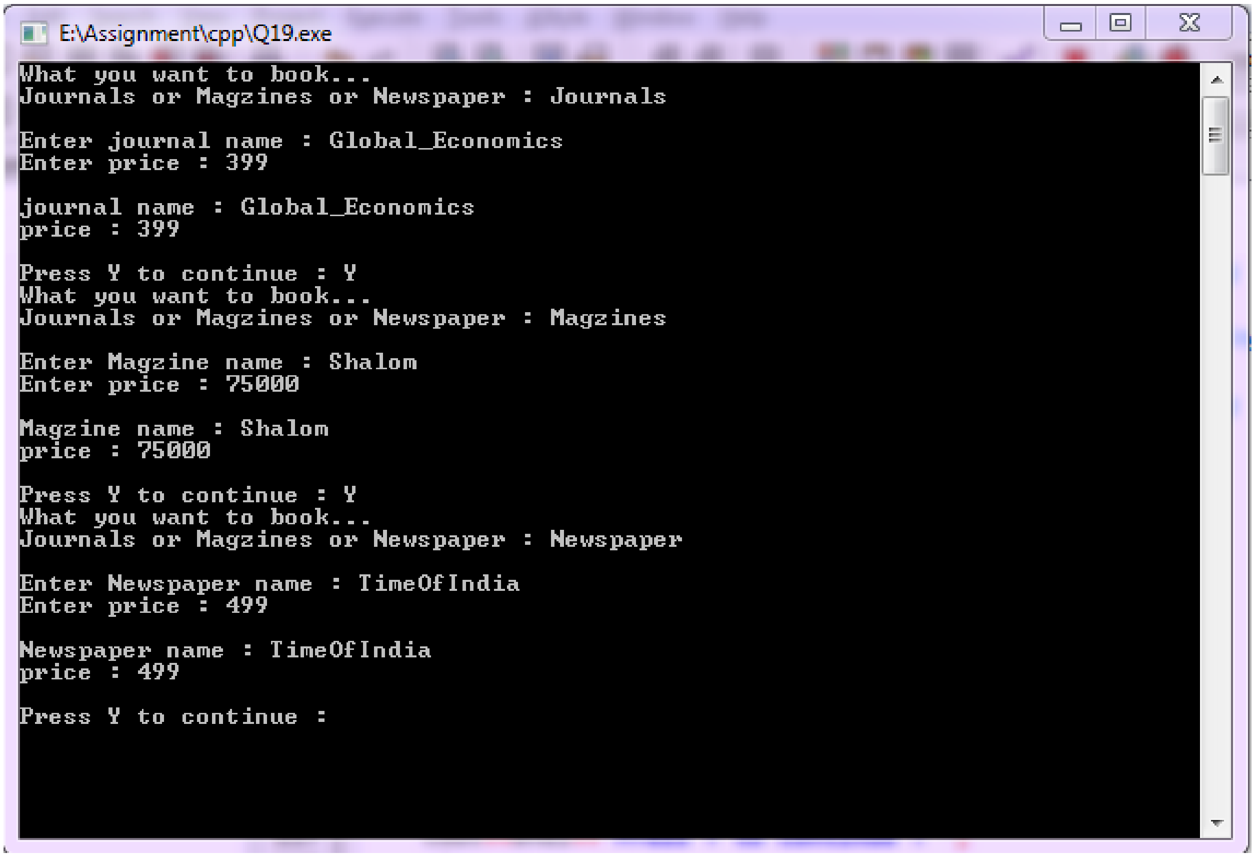
```
            char magz_name[20];
            float price;
            public:
             void get_data()
             {
               cout<<endl<<"Enter Magzine name : ";
               cin>>magz_name;
               cout<<"Enter price : ";
               cin>>price;
             }
             void disp_data()
             {       cout<<endl<<"Magzine name : "<<magz_name<<endl;
                       cout<<"price : "<<price<<endl;
             }
                                                        //member function definitions
    };
    class Newspaper                                        //base class 3 definition
    {        char news_name[20];
            float price;
            public:
             void get_data()
             {
               cout<<endl<<"Enter Newspaper name : ";
               cin>>news_name;
               cout<<"Enter price : ";
```

```
              cin>>price;
            }
            void disp_data()
            {
                    cout<<endl<<"Newspaper name : "<<news_name<<endl;
                    cout<<"price : "<<price<<endl;
            }
                                                                //member function definitions
};
class book:public Journals,public Magzines,public Newspaper
{                                               //derived class definition
                                                    //multiple inheritence
           char book_what[20];
           public:
            void booking_what()
            {
              cout<<"What you want to book... "<<endl;
              cout<<"Journals or Magzines or Newspaper : ";
              cin>>book_what;
            }
            void get_book_data()
            {
                  if(book_what == "Journals")
                    Journals::get_data();                             //fun. overriding
                    else if(book_what == "Magzines")
```

```
                        Magzines::get_data();                                    //fun. overriding
                        else if(book_what == "Newspaper" )
                        Newspaper::get_data();                                   //fun. overriding
                }
                void display_booked()
                {
                        if(book_what == "Journals")
                        Journals::disp_data();                       //fun. overriding
                        else if(book_what == "Magzines")
                        Magzines::disp_data();                       //fun. overriding
                        else if(book_what == "Newspaper")
                        Newspaper::disp_data();                      //fun. overriding
                }

};
int main()
{
        book obj;                                                //Object of derived class
        char op;
        do{
        obj.booking_what();
        obj.get_book_data();
        obj.display_booked();  //accessing own mwmber function
        cout<<endl<<"Press Y to continue : ";
        cin>>op;
```

```
  }while( op == 'y' || op == 'Y');


  getch();

  return 0;

}
```

OUTPUT :



```
What you want to book...
Journals or Magzines or Newspaper : Journals

Enter journal name : Global_Economics
Enter price : 399

journal name : Global_Economics
price : 399

Press Y to continue : Y
What you want to book...
Journals or Magzines or Newspaper : Magzines

Enter Magzine name : Shalom
Enter price : 75000

Magzine name : Shalom
price : 75000

Press Y to continue : Y
What you want to book...
Journals or Magzines or Newspaper : Newspaper

Enter Newspaper name : TimeOfIndia
Enter price : 499

Newspaper name : TimeOfIndia
price : 499

Press Y to continue :
```

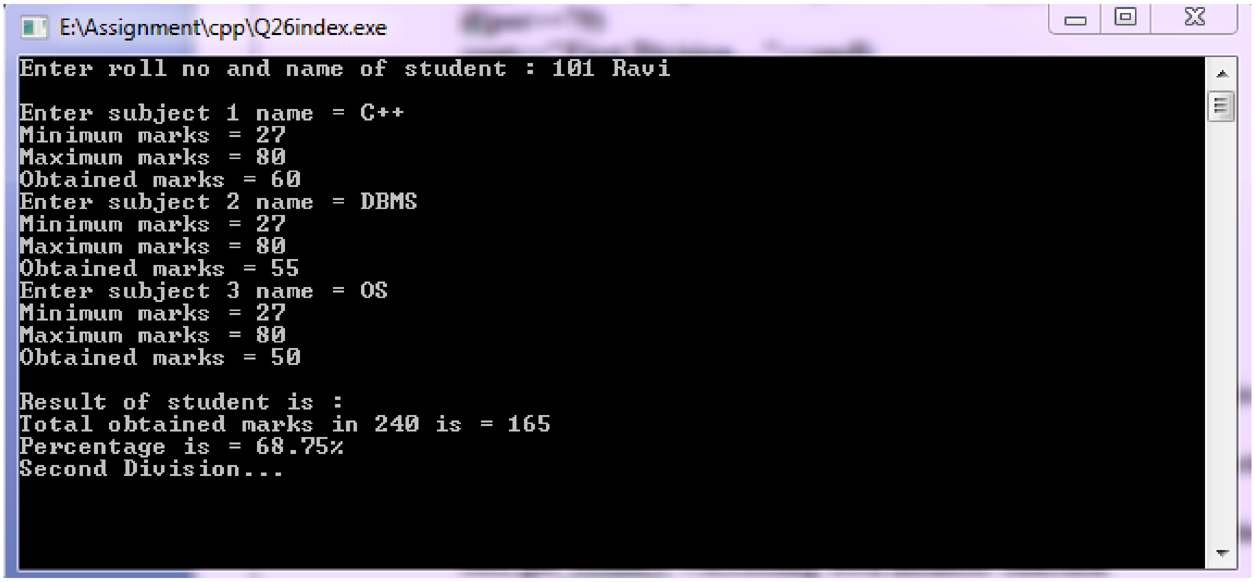
Q26. Consider an example of declaring the examination result.design 3 classes student,exam,result. The student class has data members such as that reperesenting number, name of student ,create the class exam,which contains data members reperesenting name of subject,minmum marks,maximum marks, obtained marks for 3 subject derive class result from both student and exam classes. Test the result class in main function ?

PROGRAM :

//Program to demostrate three classes(student , exam and result) which is in multiple inheritance

```
#include<iostream>
#include<conio.h>
using namespace std;
class student                                           //base class 1
{
        int rollno;
        char  name[20];                                      //Data members
        public:
        void get_student_data()
        {       cout<<"Enter roll no and name of student : ";
                cin>>rollno>>name;
        }
        void show_student_data()
        {
                cout<<"Roll no = "<<rollno<<"\t Name = "<<name<<endl;
        }
                                          //Member function declaration
};
class exam                                          // base class 2
{
        protected:
        char sub_name[3][20];
        float min_marks[3],max_marks[3],obt_marks[3];
                                                //Data members in protected mode
        public:
      void get_exam_data();
      void display_exam_data();
                                                //Member function declaration
};
class result:public student,public exam          //multiple inheritence
{
        float total_max;
        float total_obt,per;
                                                //Data members
        public:
                result()                                //Default constructor
                { total_max = 0;
                  total_obt = 0;
                  per = 0;              }
                void get_result();
                void display_result();
                                                        //Member function declaration
};
void exam::get_exam_data()                      //Member function definition of exam
{       int i;
        for(i=0; i<3; i++)
        {
                cout<<"Enter subject "<<i+1<<" name = ";
                cin>>sub_name[i];
                cout<<"Minimum marks = ";
                cin>>min_marks[i];
                cout<<"Maximum marks = ";
                cin>>max_marks[i];
                cout<<"Obtained marks = ";
                cin>>obt_marks[i];
                }
}
void exam:: display_exam_data()                 //Member function definition of exam
{
  int i;
  for(i=0; i<3; i++)
        {
        cout<<"Subject "<<i+1<<" name = "<<sub_name[i]<<endl;
        cout<<"Minimum marks = "<<min_marks[i]<<endl;
   cout<<"Maximum marks = "<<max_marks[i]<<endl;
   cout<<"Obtained marks = "<<obt_marks[i]<<endl;
        }
}
void result::get_result()                       //Member function definition of result
{
        int i;
          for(i=0; i<3; i++)
                 {
                  total_max = total_max + max_marks[i];
                        total_obt = total_obt + obt_marks[i];
            }
            per = total_obt * 100 / total_max;
}
void result::display_result()                   //Member function definition of result
{
        cout<<"Result of student is :"<<endl;
        cout<<"Total obtained marks in "<<total_max<<" is = "<<total_obt<<endl;
        cout<<"Percentage is = "<<per<<"%"<<endl;
        if(per>=70)
        cout<<"First Division...."<<endl;
        else if(per<70 && per>=50)
        cout<<"Second Division..."<<endl;
        else if(per<50 && per>=33)
        cout<<"Third Division..."<<endl;
        else
        cout<<"Fail..."<<endl;
}
int main()
{
        result res1;            //Object created and initialized of type(result) derived class

        res1.get_student_data();        //Accessing base class 1 member function
        cout<<endl;
        res1.get_exam_data();           //Accessing base class 2 member function
        res1.get_result();              //Accessing own member function
        cout<<endl;
        res1.display_result();
        getch();
        return 0;
}
```

OUTPUT :



```
Enter roll no and name of student : 101 Ravi

Enter subject 1 name = C++
Minimum marks = 27
Maximum marks = 80
Obtained marks = 60
Enter subject 2 name = DBMS
Minimum marks = 27
Maximum marks = 80
Obtained marks = 55
Enter subject 3 name = OS
Minimum marks = 27
Maximum marks = 80
Obtained marks = 50

Result of student is :
Total obtained marks in 240 is = 165
Percentage is = 68.75%
Second Division...
```

Q27. WAP to generate fibbonacci series use the concept of function overriding.

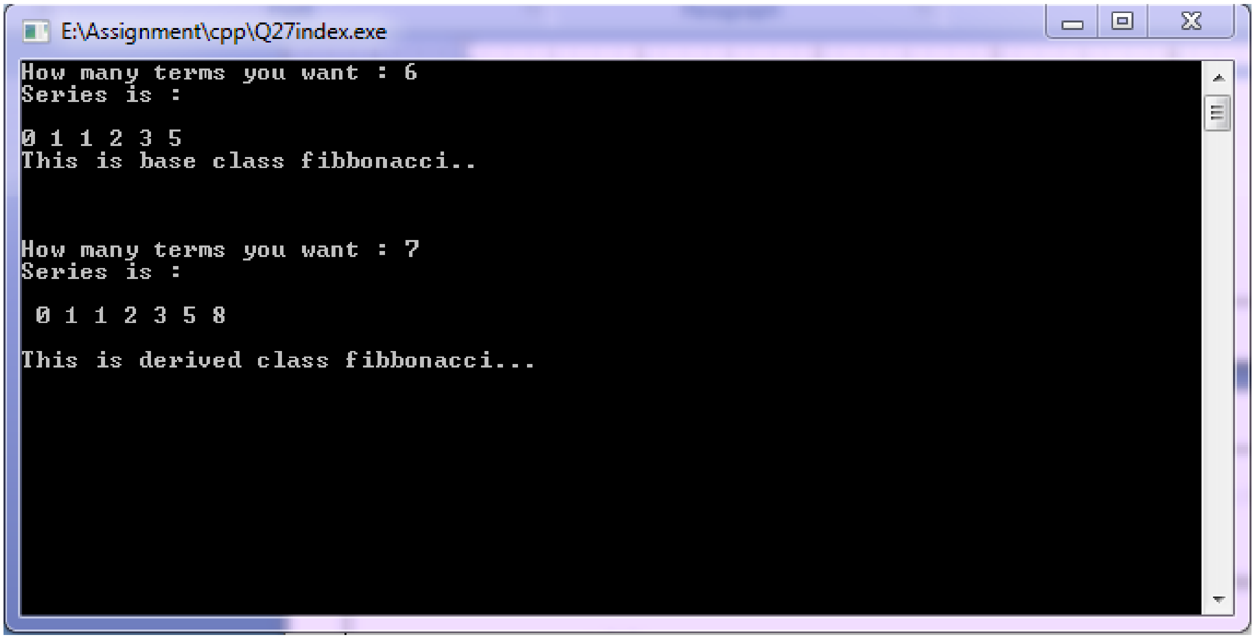
PROGRAM :

```
                    //Program for fibbonacci series using function overriding
#include<iostream>
#include<conio.h>
using namespace std;
class base   //base class
{     public:
   void fibbo(int);                                    //member fun. declaration
};
void base::fibbo(int n)                                //member fun. definition
{        int n1=0,n2=1,n3;
         int i;
         cout<<"Series is : "<<endl<<endl;
         cout<<n1<<" "<<n2;
         for(i=0; i<n-2; i++)
         {        n3=n1+n2;
                  cout<<" "<<n3;
                  n1 = n2;
                  n2 = n3;     }
                  cout<<endl<<"This is base class fibbonacci.."<<endl<<endl;
}
```

```
class derived:public base                    //publically derivation of base class
{
        public:
        int fibbo(int n)                     //same name as base class function
        {       if(n==0 || n==1)
                return n;
                else
                return fibbo(n-1)+fibbo(n-2);
         }                                            //member fun. Definition
 };
int main()
{       derived d;                                            //object of derived class
        int n;
        cout<<"How many terms you want : ";
        cin>>n;
        d.base::fibbo(n);                                     //function overriding
   int i=0;
   cout<<endl<<endl;
   cout<<"How many terms you want : ";
        cin>>n;
        cout<<"Series is :"<<endl<<endl;
        while(i<n){
        cout<<" "<<d.derived::fibbo(i);                               //function overriding
```

```
            i++;

            }

             cout<<endl<<endl<<"This is derived class fibbonacci..."<<endl;

             getch();

            return 0;

}
```

OUTPUT :



```
How many terms you want : 6
Series is :

0 1 1 2 3 5
This is base class fibbonacci..



How many terms you want : 7
Series is :

 0 1 1 2 3 5 8

This is derived class fibbonacci...
```

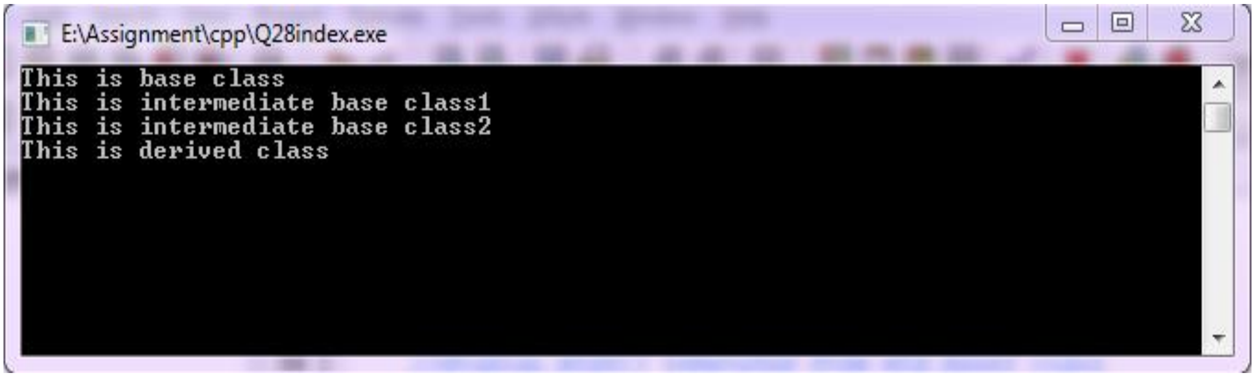
Q28. Write a program to solve Diamond problem(Hybrid inheritance and virtual base class).

PROGRAM :

//Program to solve Diomand problem

```
#include<iostream>
#include<conio.h>
using namespace std;
class base                                              //base class
{
        public:
                void display_base()
                {
                        cout<<"This is base class"<<endl;
                }
};
class mid_base1: virtual public base                    //virtual base class
{
        public:
            //display_base() inherited from base class
                void display_mid1()
                {
                        cout<<"This is intermediate base class1"<<endl;
                }
};
class mid_base2:public virtual base //vitual base class
{
        public:
                //display_base() inherited from base class
                void display_mid2()
                {
                        cout<<"This is intermediate base class2"<<endl;
                }
};

class derived:public mid_base1,public mid_base2
{
        public:
            //display_mid1() inheruted from mid_base1 class
            //display_base() inherited from mid_base2 class
    //which display_base() should be inherited from mid_base1 or from mid_base2
    //problem solved because base class is virtual otherwise it shows ambigiuty

void display_derived()
                        {
                                    cout<<"This is derived class"<<endl;
                        }
};
int main()
{
            derived d;                              //created object of derived class
            d.display_base();
            d.display_mid1();
            d.display_mid2();
                                                    //accessing inherited member functions
            d.display_derived();
            getch();
            return 0;
}
```

OUTPUT :



E:\Assignment\cpp\Q28index.exe

```
This is base class
This is intermediate base class1
This is intermediate base class2
This is derived class
```

Q29. Write a program in c++ using constructor and destructor in Multiple and multilevel inheritance.

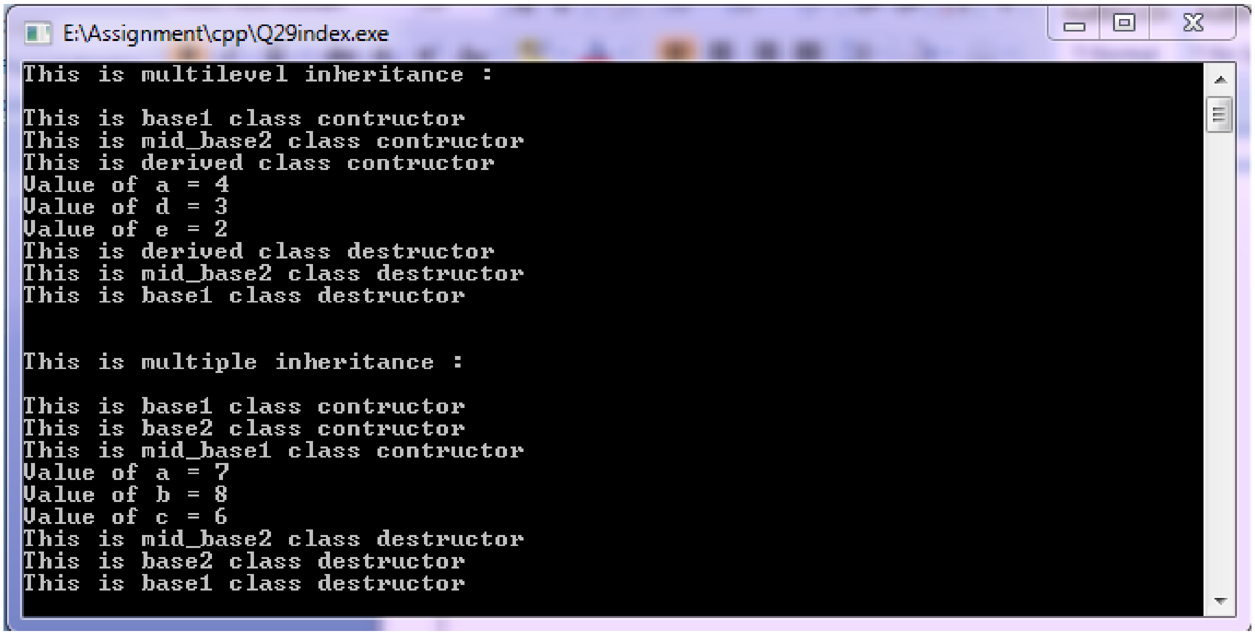
PROGRAM :

```
                    //Program to demostrate multiple and multilevel inheritance

#include<iostream>
#include<conio.h>
using namespace std;

class base1
{
        int a;
        public:
                base1(int a1)                                   //constructor of base class
                {
                        a = a1;
                        cout<<"This is base1 class contructor"<<endl;
                }
                ~base1()                //destructor
                {
                        cout<<"This is base1 class destructor"<<endl;
                }
                void display()
                {
                        cout<<"Value of a = "<<a<<endl;
                }
};

class base2
{
        int b;
        public:
                base2(int b1)                   //contrutor of base class
                {
                        b = b1;
                        cout<<"This is base2 class contructor"<<endl;
                }
                ~base2()                        //destructor
                {
                        cout<<"This is base2 class destructor"<<endl;
                }
                void display()
                {
                        cout<<"Value of b = "<<b<<endl;
                }
};

class mid_base1 : public base1,public base2 //multiple inheritance
{
        int c;
        public:
        mid_base1(int c1,int c2,int c3):base1(c2),base2(c3)
        {                       //base classes contructor calling(multiple inheri.)
                                  //3 argumnts passed to mid_base1
                                   //c2 passed to base1 and c3 passed to base2
                        c = c1;
                        cout<<"This is mid_base1 class contructor"<<endl;
          }
                ~mid_base1()      //destructor
                {
                        cout<<"This is mid_base2 class destructor"<<endl;
                }
                void display()
                {
                        cout<<"Value of c = "<<c<<endl;
                }
};

class mid_base2:public base1  //mid_base2 is derived from base1 class
{
        int d;
        public:
        mid_base2(int d1,int d2):base1(d2)  //calling constructor of base of mid_base2 class
                {                    //2 arguments passed to mid_base2
                                    //d2 passed to class base1
                        d = d1;
                        cout<<"This is mid_base2 class contructor"<<endl;
                }
                ~mid_base2()            //destructor
                {
                        cout<<"This is mid_base2 class destructor"<<endl;
                }
                void display()
                {
```

```
                cout<<"Value of d = "<<d<<endl;
            }
};

class derived: public mid_base2                     //multilevel inheritance
{
        int e;
        public:
            derived(int e1,int e2,int e3):mid_base2(e2,e3)
                                   //base classes contrutors calling(multilevel)
            {               //Passing 3 arguments to derived class
                             //e2 and e3 passed to class mid_base2
                e = e1;
                cout<<"This is derived class contructor"<<endl;
            }
            ~derived()              //destructor
            {
                cout<<"This is derived class destructor"<<endl;
            }
            void display()
            {
                cout<<"Value of e = "<<e<<endl;
            }
};

int main()
{
        cout<<"This is multilevel inheritance : "<<endl<<endl;
        {
                //multileval inheritance
        derived d(2,3,4);                   //passing three agrument to derivered class object
        d.base1::display();             //fun. overriding
        d.mid_base2::display();
        d.derived ::display();
  }
        cout<<endl<<endl;
         cout<<"This is multiple inheritance : "<<endl<<endl;
        {                               //multiple inheritance
        mid_base1 m(6,7,8);              //passing three agrument to derivered class object
        m.base1::display();
        m.base2::display();
        m.mid_base1::display();
                  //funcvtion overriding
  }
        getch();
        return 0;
```

}

OUTPUT :



E:\Assignment\cpp\Q29index.exe

```
This is multilevel inheritance :

This is base1 class contructor
This is mid_base2 class contructor
This is derived class contructor
Value of a = 4
Value of d = 3
Value of e = 2
This is derived class destructor
This is mid_base2 class destructor
This is base1 class destructor


This is multiple inheritance :

This is base1 class contructor
This is base2 class contructor
This is mid_base1 class contructor
Value of a = 7
Value of b = 8
Value of c = 6
This is mid_base2 class destructor
This is base2 class destructor
This is base1 class destructor
```

Q30. Write a program in c++ to demonstrate pointer to an object and this pointer.

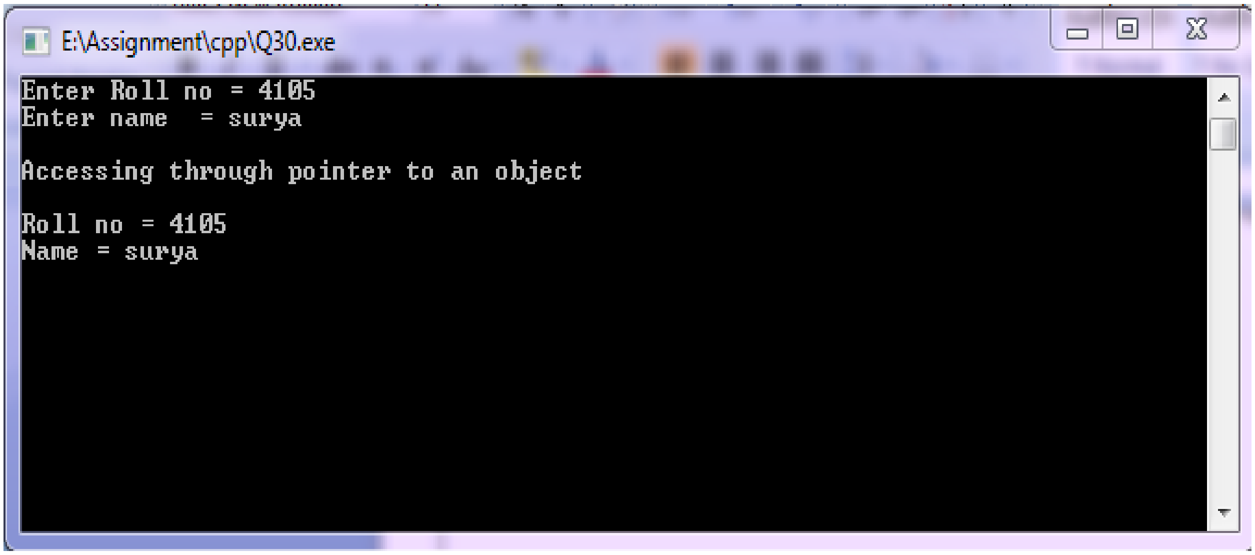
PROGRAM :

//Program to demonstrate pointer to an object and this pointer

```
#include<iostream>
#include<conio.h>
using namespace std;
class student
{
        int rollno;
        char name[20];
                                                    //Data members
   public:
         student() {      }                         //Default constructor

     student(int rollno,char name[20])              //Parameterized constructor
     {
      this->rollno = rollno;                        //this pointer stores address of caller object
      this->name = name;
      }

   void display()                                   //member function definition
   {
         cout<<endl<<"Roll no = "<<rollno<<endl;
         cout<<"Name = "<<name<<endl;
   }
};
int main()
{
        int rollno1;
        char name1[20];

        cout<<"Enter Roll no = ";
        cin>>rollno1;
        cout<<"Enter name  = ";
        cin>>name1;
        student s(rollno1,name1);
```

```
                                          //object created and parameterized constructor invoked
        student *ptr;                     //pointer of type student



        ptr = &s;
                              //Pointer to an object

        cout<<endl<<"Accessing through pointer to an object "<<endl;
        ptr->display();
                           //Accessing member function through pointer to an object

        getch();
        return 0;
}
```

OUTPUT :



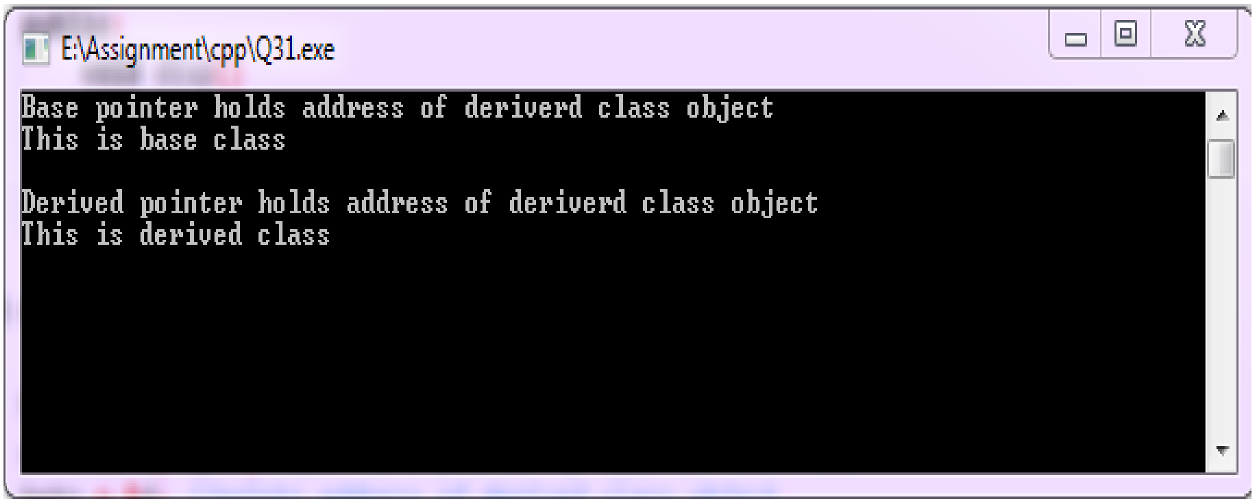
```
Enter Roll no = 4105
Enter name  = surya

Accessing through pointer to an object

Roll no = 4105
Name = surya
```

Q31. Write a program in c++ to demonstrate pointer to derived class.

PROGRAM :

```
                              //Program to demostrate pointer to derived class

#include<iostream>
#include<conio.h>
using namespace std;
class base
{
        public:
                void disp()
                {
                        cout<<"This is base class"<<endl;
                }

};
class derived:public base  //single imheritance
{
        public:
                void disp()
                {
                        cout<<"This is derived class"<<endl;
                }
};
int main()
{
        base *bptr,b;                           //bptr is a pointer of type base class
        derived d,*dptr;                        //dptr is a pointer of type derived class
        bptr = &d;                              //points address of derived class object
        cout<<"Base pointer holds address of deriverd class object "<<endl;
        bptr->disp();
//disp() of base class cause of bptr = &d is just ignoured by compiler at compile time

        cout<<endl;
        dptr = &d;                              //points address of own class object
        cout<<"Derived pointer holds address of deriverd class object "<<endl;
        dptr->disp();                                   //disp() of derived class
        getch();
        return 0;
}
```

OUTPUT :



```
Base pointer holds address of deriverd class object
This is base class

Derived pointer holds address of deriverd class object
This is derived class
```

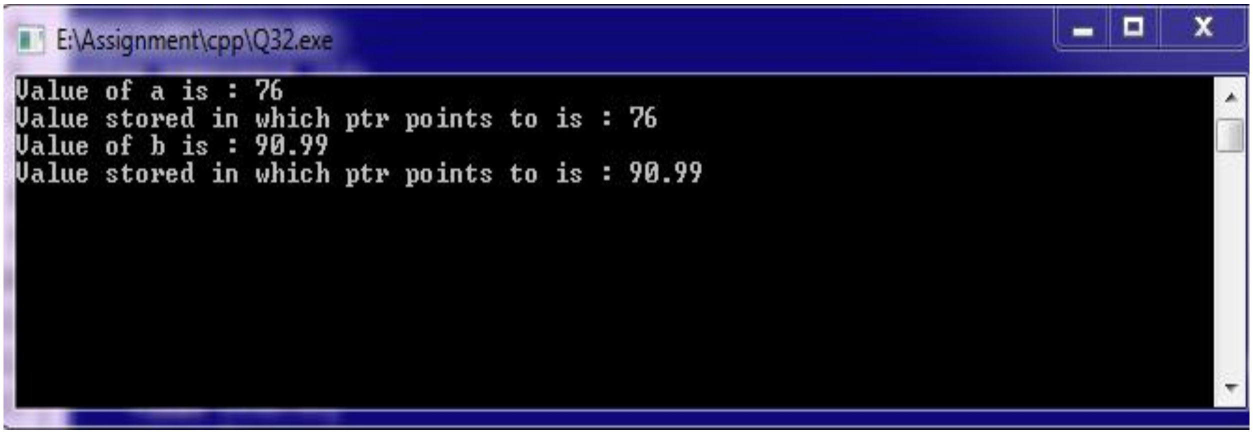
Q32. Create a program having pointer to void to store address of integer variable then print value of integer variable using pointer to void. Perform the same operation for float variable.

PROGRAM :

//Program to demonstrate void pointer

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
        void *ptr;                                                    //void pointer
        int a=76;
   ptr = &a;                                                    //ptr holds address of a
   cout<<"Value of a is : "<<a<<endl;
   cout<<"Value stored in which ptr points to is : "<<*(int*)ptr<<endl;
                                                        //type casting of void pointer to int
   float b=90.99;
   ptr = &b;
   cout<<"Value of b is : "<<b<<endl;
   cout<<"Value stored in which ptr points to is : "<<*(float*)ptr<<endl;
                                                        //type casting of void pointer to float
        getch();
        return 0;
}
```

OUTPUT :



```
Value of a is : 76
Value stored in which ptr points to is : 76
Value of b is : 90.99
Value stored in which ptr points to is : 90.99
```

Q33. Create a class account that stores customer name, account number and type of account. From this derive the classes cur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks:

a) Accept deposit from customer.

b) Display the balance

PROGRAM :

//Program to accept deposit and display the balance of saving or current account

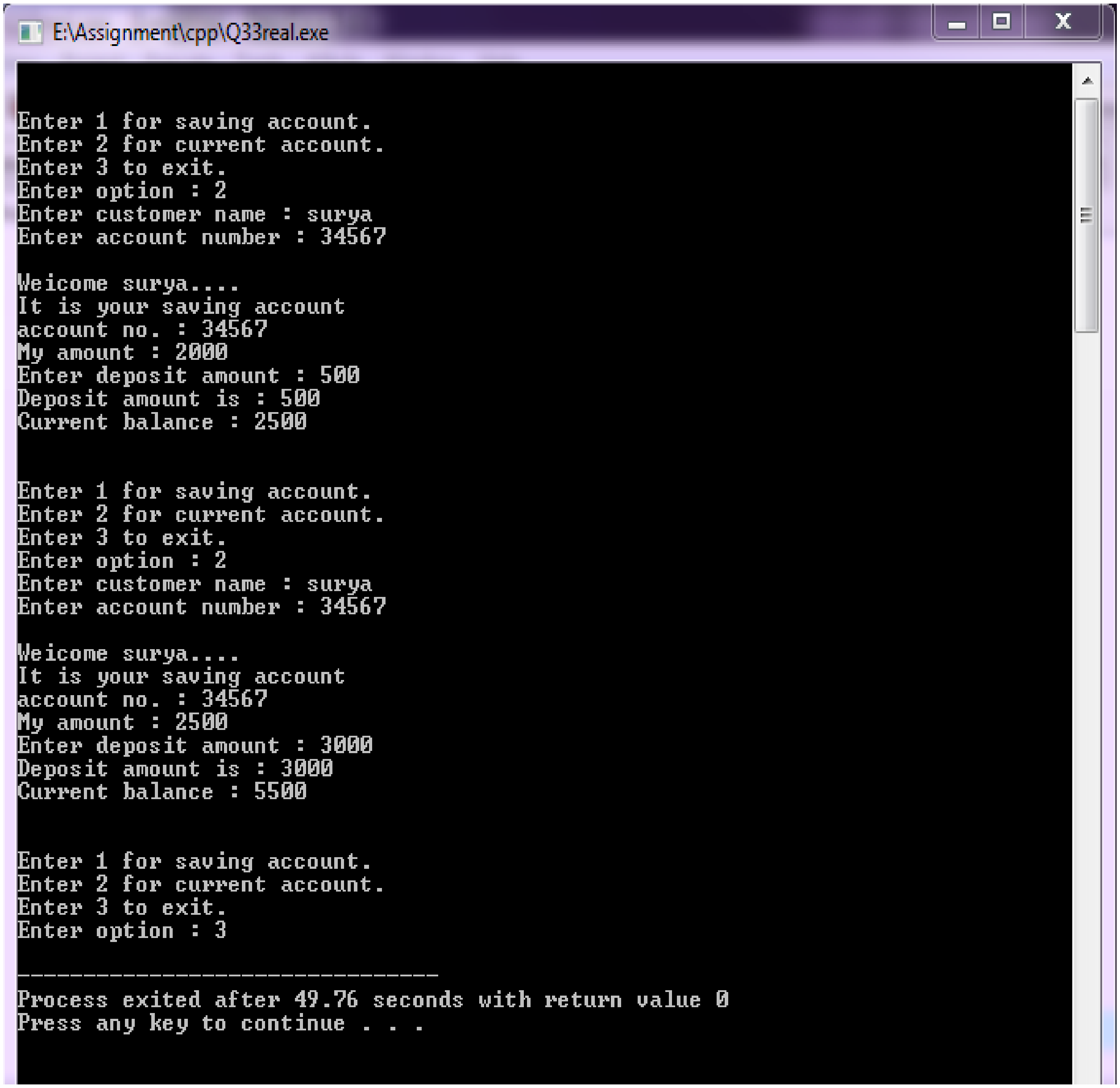
```
#include<iostream>
#include<conio.h>
using namespace std;
class Account                                   //base class
{
protected:
   char cust_name[30],acct_type[30];
   int account_number;
   float amount;
public:
         Account()                              //default constructor
         {
         amount=2000;
         }
   void get_detail()                            //member function definition
   {
         cout<<"Enter customer name : ";
         cin>>cust_name;
         cout<<"Enter account number : ";
         cin>>account_number;
   }
```

```
};

class cur_acct:public Account                          //derived class
{
float deposit;

public:
void current();
void get_deposit();
void show_deposit();                           //member fun. declaration
};
void cur_acct::current()                       //member fun. definition
{
get_detail();
cout<<"\nWeicome "<<cust_name<<"...."<<endl;
cout<<"It is your current account"<<endl;
cout<<"account no. : "<<account_number<<endl;
cout<<"my amount : "<<amount<<endl;
}
void cur_acct:: get_deposit()
{
cout<<"enter deposit amount : ";
cin>>deposit;
amount=amount+deposit;
}
void cur_acct:: show_deposit()
{
cout<<"deposit amount is : "<<deposit<<endl;
cout<<"current balance : "<<amount<<endl;
}

class sav_acct:public Account
{
float deposit;

public:
void saving();
void get_deposit();
void show_deposit();                      //member fun. declaration
};
void sav_acct::saving()                   //member fun. defintion
{
get_detail();
cout<<"\nWeicome "<<cust_name<<"...."<<endl;
cout<<"It is your saving account"<<endl;
cout<<"account no. : "<<account_number<<endl;
```

```
cout<<"My amount : "<<amount<<endl;
}
void sav_acct:: get_deposit()
{
cout<<"Enter deposit amount : ";
cin>>deposit;
amount=amount+deposit;
}
void sav_acct:: show_deposit()
{
cout<<"Deposit amount is : "<<deposit<<endl;
cout<<"Current balance : "<<amount<<endl;
}

int main()
{
int i;
cur_acct obj1;
sav_acct obj2;          //objects of derived class
do{
cout<<endl<<endl;
cout<<"Enter 1 for saving account."<<endl;
cout<<"Enter 2 for current account."<<endl;
cout<<"Enter 3 to exit."<<endl;
cout<<"Enter option : ";
cin>>i;
        switch(i)
        {
        case 1:
           obj1.current();
           obj1.get_deposit();
           obj1.show_deposit();
           break;
        case 2:
           obj2.saving();
           obj2.get_deposit();
           obj2.show_deposit();
           break;
        case 3:
           exit(0);
        default:
           cout<<"Enter valid option..."<<endl;
        }
}while(1);
 getch();
 return 0;
```

```
}
```

OUTPUT :



```

Enter 1 for saving account.
Enter 2 for current account.
Enter 3 to exit.
Enter option : 2
Enter customer name : surya
Enter account number : 34567

Weicome surya....
It is your saving account
account no. : 34567
My amount : 2000
Enter deposit amount : 500
Deposit amount is : 500
Current balance : 2500


Enter 1 for saving account.
Enter 2 for current account.
Enter 3 to exit.
Enter option : 2
Enter customer name : surya
Enter account number : 34567

Weicome surya....
It is your saving account
account no. : 34567
My amount : 2500
Enter deposit amount : 3000
Deposit amount is : 3000
Current balance : 5500


Enter 1 for saving account.
Enter 2 for current account.
Enter 3 to exit.
Enter option : 3

--------------------------------
Process exited after 49.76 seconds with return value 0
Press any key to continue . . .
```

Q34. Create a class circle with data member radius; provide member function to calculate area. Derive a class sphere from class circle; provide member function to calculate

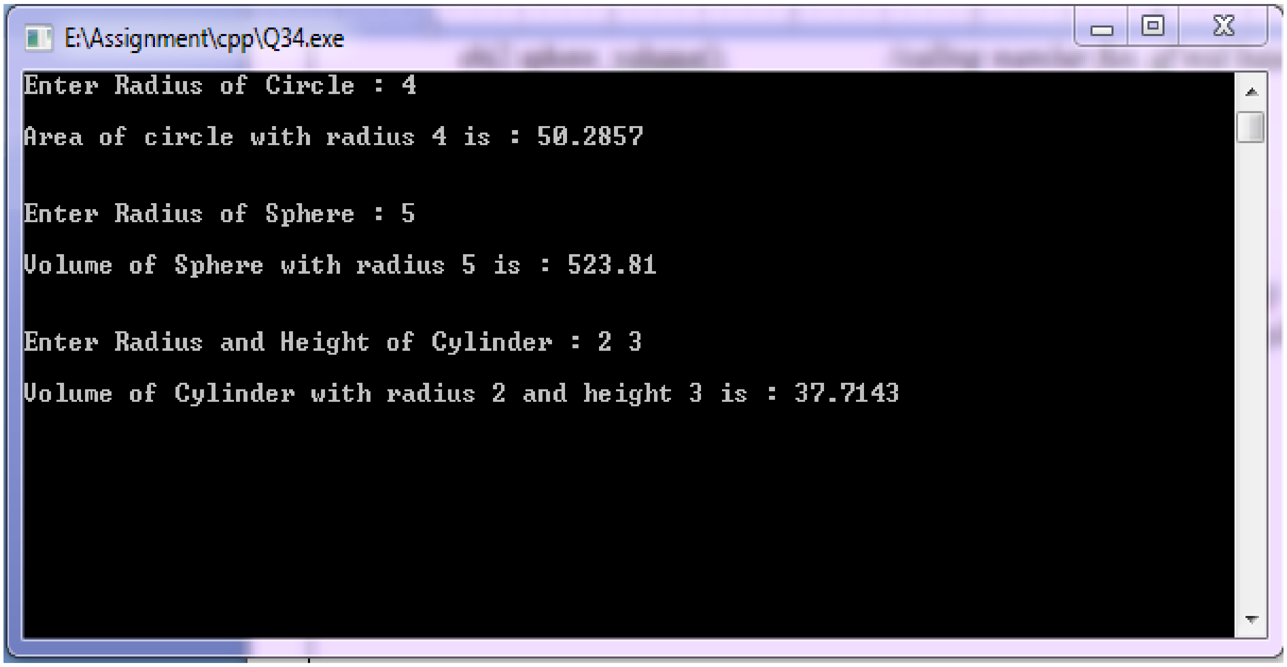
volume.Derive class cylinder from class sphere with additional data member for height and member function to calculate volume.

PROGRAM :

//Program to calculate volume of cylinder and sphere and area of circle through multilevel inheritance

```
#include<iostream>
#include<conio.h>
using namespace std;
const float pi=22/7.0;                              //constant variable pi
class circle                          //base class
{
        protected:
                float radius;
                                      //Data member in protected mode
        public:
                circle(float radius)           //parameterized constructor
                {
                        this->radius = radius;
                }
                void circle_area()             //member function definition
                {
        cout<<"Area of circle with radius "<<radius<<" is : "<<(pi) * radius * radius<<endl;
                }

};
class sphere:public circle            //intermediate base class
{
        public:
                sphere(float r):circle(r)      //constructor calling statement for base class
                {

                }
                void sphere_volume()           //member function defintion
                {
                        float volume = (4.0/3)*(pi)*radius*radius*radius;
        cout<<"Volume of Sphere with radius "<<radius<<" is : "<<volume<<endl;
                }
};



class cylinder:public sphere                   //derived class (multilevel inheritance)
{
        float height;                           //data member
public:
        cylinder(float r,float height):sphere(r)
                        //r passed to base class and height initialized with data member height
        {
                this->height = height;
        }
        cylinder(float r):sphere(r)     //only radius is passed to base class
        {
                height=0;
        }
        void cylinder_volume()                  //member function definition
        {
          float volume = pi * radius * radius * height;
          cout<<"Volume of Cylinder with radius "<<radius<<" and height "<<height<<" is :
          "<<volume<<endl;
        }
};
int main()
{
   float r;
        cout<<"Enter Radius of Circle : ";
        cin>>r;
        cout<<endl;
   cylinder obj1(r);            //constructor cylinder(float) invoked
                //object of derived(cylinder) class created and radius passed to base class(circle)
   obj1.circle_area();                  //calling member fun. of base (circle) class
        cout<<endl<<endl;
        cout<<"Enter Radius of Sphere : ";
        cin>>r;
        cout<<endl;
   cylinder obj2(r);                    //constructor cylinder(float) invoked
   obj2.sphere_volume();                //calling member fun. of mid base (sphere) class
        cout<<endl<<endl;
   float h;
        cout<<"Enter Radius and Height of Cylinder : ";
        cin>>r>>h;
        cout<<endl;
   cylinder obj3(r,h);                  //constructor cylinder(float,float) invoked
   obj3.cylinder_volume();              //calling member fun. of own (cylinder) class
        getch();
        return 0;
}
```

OUTPUT :



```
Enter Radius of Circle : 4

Area of circle with radius 4 is : 50.2857


Enter Radius of Sphere : 5

Volume of Sphere with radius 5 is : 523.81


Enter Radius and Height of Cylinder : 2 3

Volume of Cylinder with radius 2 and height 3 is : 37.7143
```

Q35. Write a program in c++ for overloading of unary operator.

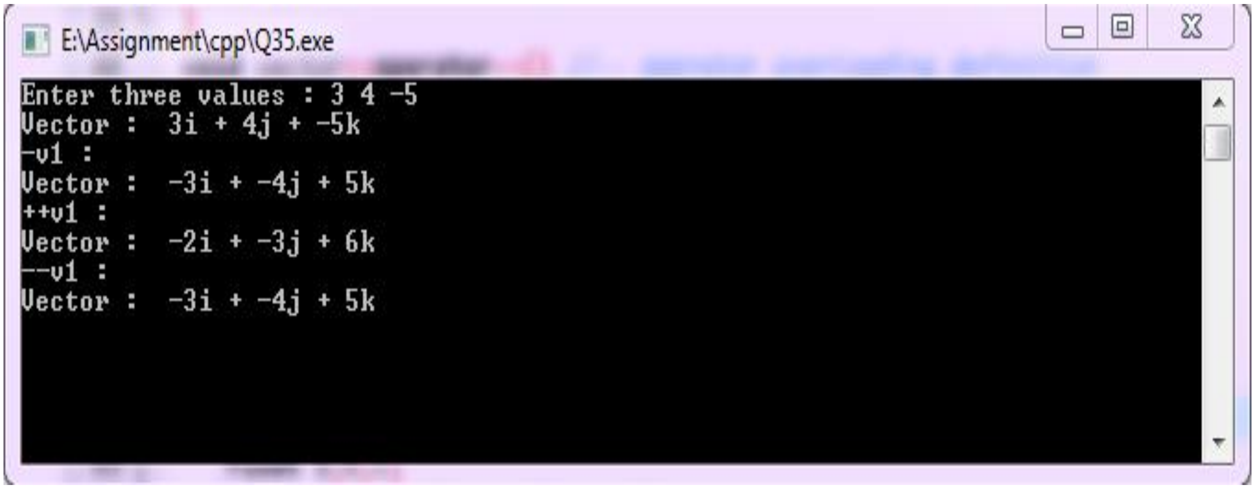
PROGRAM :

//Program to overload unary operator

```
#include<iostream>
#include<conio.h>
using namespace std;
class vector
{
  float x;
  float y;
  float z;  //data members
    public:
            vector(){   }                         //default constructor
             vector(float x,float y,float z)                        //parameterized constructor
             {
                     this->x = x;
                     this->y = y;
                     this->z = z;
             }
             void display();                    //member functions declaration
             void operator-();                  //- operator overloading declaration
             void operator++();                 //++ operator overloading declaration
             void operator--();                  //-- operator overloading declaration
};
void vector::display()                          //member function definition
{
            cout<<"Vector :  "<<x<<"i + "<<y<<"j + "<<z<<"k"<<endl;

}
void vector::operator-()                        //- operator overloading definition
{
                     x = -x;
                     y = -y;
                     z = -z;
 }
 void vector::operator++()                       //++ operator overloading definition
 {
            ++x;
            ++y;
            ++z;
 }


 void vector::operator--()                              //-- operator overloading definition
```

```
{
        --x;
        --y;
        --z;
}
int main()
{
        float a,b,c;
        cout<<"Enter three values : ";
        cin>>a>>b>>c;
        vector v1(a,b,c);              //object created and parameterized constructor invoked
        v1.display();                  //member function calling
        cout<<"-v1 : "<<endl;
        -v1;                           //operator- function calling
        v1.display();
        cout<<"++v1 : "<<endl;
        ++v1;                          //operator++ function calling
        v1.display();
        cout<<"--v1 : "<<endl;
        --v1;                           //operator-- function calling
        v1.display();
        getch();
        return 0;
}
```

OUTPUT :



```
Enter three values : 3 4 -5
Vector :  3i + 4j + -5k
-v1 :
Vector :  -3i + -4j + 5k
++v1 :
Vector :  -2i + -3j + 6k
--v1 :
Vector :  -3i + -4j + 5k
```

Q36. Write a program in c++ for overloading of binary operator.

PROGRAM :

//Program for overloading binary operators

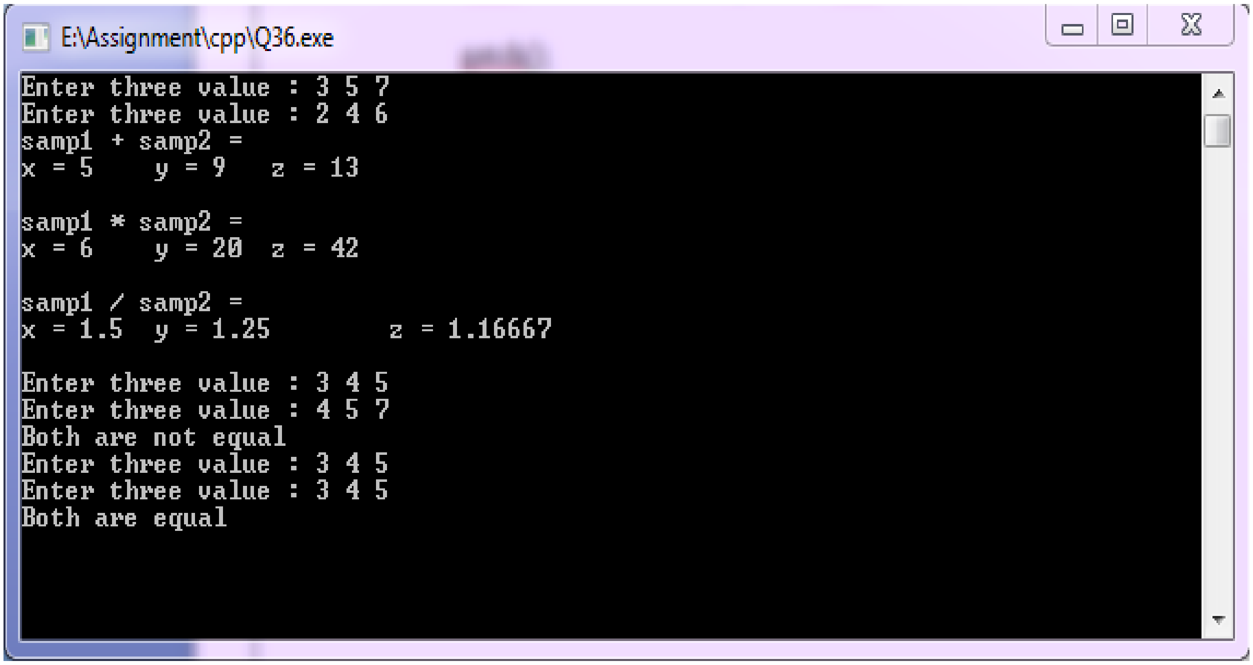
```
#include<iostream>
#include<conio.h>
using namespace std;
class sample  //class definition
{       float x;
        float y;
        float z;                                        //data members
          public:
        void get_sample();                              //member function declaretion
           //operator overloading
        sample operator+(sample);          //operator+(binary) member function declaration
        sample operator*(sample);          //operator*(binary) member function declaration
        sample operator/(sample);          //operator/ (binary) member function declaration
        bool operator==(sample);           //operator== (binary) member function declaration
        void disp_sample();                //member functiom declaration

};
void sample::get_sample()                  //member functiom definition
{ cout<<"Enter three value : ";
  cin>>x>>y>>z;        }
void sample::disp_sample()                          //member functiom declaration
```

```
{ cout<<"x = "<<x<<"\t y = "<<y<<"\t z = "<<z<<endl<<endl; }
sample sample::operator+(sample s)          //operator+(binary) member function definition
{       sample temp;
        temp.x = x + s.x;
        temp.y = y + s.y;
        temp.z = z + s.z;
        return temp;
}
sample sample::operator*(sample s)          //operator* (binary) member function definition
{       sample temp;
        temp.x = x * s.x;
        temp.y = y * s.y;
        temp.z = z * s.z;
        return temp;
}
sample sample::operator/(sample s)   //operator/ (binary) member function definition
{       sample temp;
        temp.x = x / s.x;
        temp.y = y / s.y;
        temp.z = z / s.z;
        return temp;    }
bool sample::operator==(sample s)           //operator== (binary) member function definition
{        if(x==s.x && y==s.y && z==s.z)
        return true;
        else
```

```
            return false;
    }
    int main()
    {
            sample samp1,samp2,samp3;
            samp1.get_sample();
            samp2.get_sample();
            cout<<"samp1 + samp2 = "<<endl;
            samp3=samp1+samp2;                          //calling operator+ function
                  //same as samp3 = samp1.operator+(samp2);
            samp3.disp_sample();
            cout<<"samp1 * samp2 = "<<endl;
            samp3=samp1 * samp2;                        //calling operator* function
            samp3.disp_sample();
            cout<<"samp1 / samp2 = "<<endl;
            samp3=samp1 / samp2;                         //calling operator/ function
            samp3.disp_sample();
            samp1.get_sample();
            samp2.get_sample();
            if(samp1==samp2)                            //calling operator== function
            cout<<"Both are equal "<<endl;
             else
             cout<<"Both are not equal "<<endl;
             samp1.get_sample();
             samp2.get_sample();
```

```
        if(samp1==samp2)                    //calling operator== function
        cout<<"Both are equal "<<endl;
        else
        cout<<"Both are not equal "<<endl;
       getch();
       return 0;
}
```

OUTPUT :



```
Enter three value : 3 5 7
Enter three value : 2 4 6
samp1 + samp2 =
x = 5    y = 9   z = 13

samp1 * samp2 =
x = 6    y = 20  z = 42

samp1 / samp2 =
x = 1.5  y = 1.25        z = 1.16667

Enter three value : 3 4 5
Enter three value : 4 5 7
Both are not equal
Enter three value : 3 4 5
Enter three value : 3 4 5
Both are equal
```

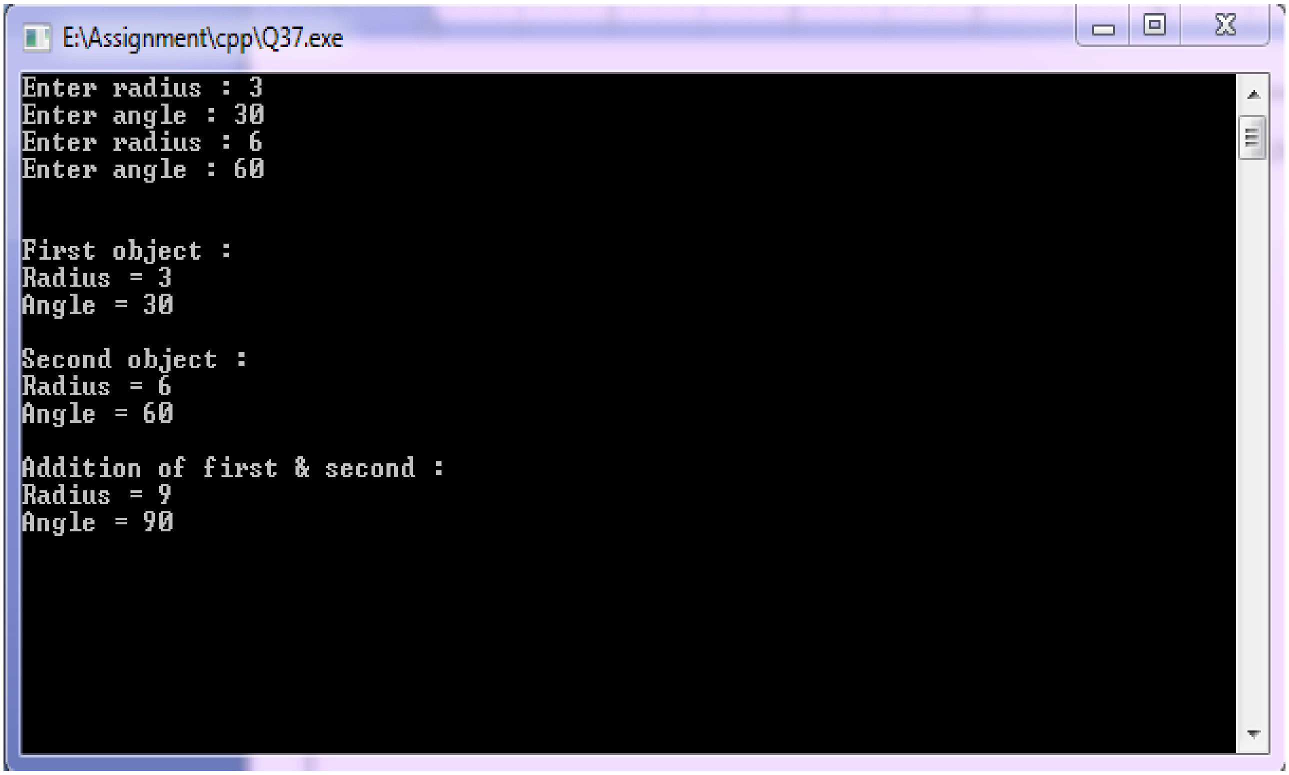
Q37. Create class Polar having data members radius and angle.It contains member functions for taking input in data member function for displaying value of data members . Class Polar contains declaration of friend function add which accepts two objects of class Polar and returns objects of class Polar after addition.Test the class using main function and object of class Polar.

PROGRAM :

```
                    //Program to add two objects of a class using friend function

#include<iostream>
#include<conio.h>
using namespace std;
class Polar
{
        float radius;
        float angle;                                        //data members
public:
         void input()
        {
                cout<<"Enter radius : ";
                cin>>radius;
                cout<<"Enter angle : ";
                cin>>angle;
        }
        void display()                                      //member fun. definition
        {
                cout<<"Radius = "<<radius<<endl;
                cout<<"Angle = "<<angle<<endl;
        }
        friend Polar add(Polar,Polar);                       //friend function declaration
};
Polar add(Polar p1,Polar p2)                                //friend function definition
{
   Polar temp;
        temp.radius = p1.radius + p2.radius;
        temp.angle = p1.angle + p2.angle;
        return temp;
                        //Adding two Polar object then return a Polar object
}
int main()
{
        Polar p1,p2,addition;                       //objects created
        p1.input();
        p2.input();
```

```
            cout<<endl;
            addition = add(p1,p2);                    //calling friend function add(Polar,Polar)
            cout<<endl<<"First object : "<<endl;
            p1.display();
            cout<<endl<<"Second object : "<<endl;
            p2.display();
            cout<<endl<<"Addition of first & second : "<<endl;
            addition.display();
            getch();
            return 0;
}
```

OUTPUT :



```
Enter radius : 3
Enter angle : 30
Enter radius : 6
Enter angle : 60


First object :
Radius = 3
Angle = 30

Second object :
Radius = 6
Angle = 60

Addition of first & second :
Radius = 9
Angle = 90
```

Q38. Write program to create a class distance having data members feet and inch (A single object will store distance in form such as 5 feet 3 inch).
It contains member functions for taking input in data members and member function for displaying value of data members.
Class Distance contains declaration of friend finction add which accepts two objcts of class distance and return objects of class Distance after addition .
Class Distance contains declaration of another friend finction Subtract that accepts two objcts of class distance and return objects of class Distance after subtraction.
Test the class using main function and objects of class Distance.

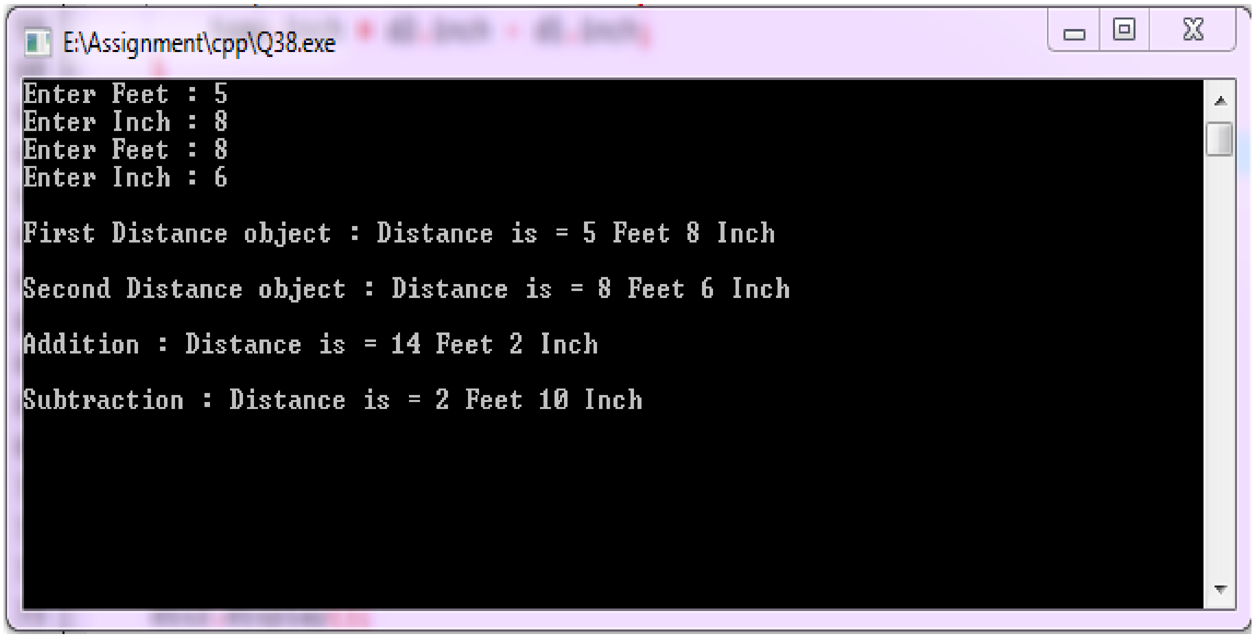
PROGRAM :

```
            //Program to add  and subtract two Distance objects using friend function

#include<iostream>
#include<conio.h>
#include<math.h>
using namespace std;
class Distance
{
        int feet;
        int inch;                                       //Data members
public:
        void input()
        {
                cout<<"Enter Feet : ";
                cin>>feet;
                cout<<"Enter Inch : ";
                cin>>inch;

        }
        void display()                                  //member fun. definition
        {
                cout<<"Distance is = "<<abs(feet)<<" Feet "<<abs(inch)<<" Inch"<<endl;
        }
        friend Distance add(Distance,Distance);
        friend Distance subtract(Distance,Distance);
                                        //Friend function declaration
};


Distance add(Distance d1,Distance d2) //friend function definition
{
        Distance temp;
        temp.feet = d1.feet + d2.feet;
```

```
            temp.inch = d1.inch + d2.inch;
            if(temp.inch>12)
                {
                        temp.feet = temp.feet + temp.inch / 12;
                        temp.inch = temp.inch % 12;
              }
            return temp;
}

Distance subtract(Distance d1,Distance d2)  //friend function definition
{
            Distance temp;

     if(d1.feet>d2.feet)
     {
            if(d1.inch<d2.inch)
            {
                         d1.feet = d1.feet - 1;
                         d1.inch = d1.inch + 12;
               }
            temp.feet = d1.feet - d2.feet;
            temp.inch = d1.inch - d2.inch;
    }
    if(d1.feet<d2.feet)
    {
            if(d1.inch>d2.inch)
            {
                        d2.feet = d2.feet - 1;
                                    d2.inch = d2.inch + 12;
                        }
                        temp.feet = d2.feet - d1.feet;
                        temp.inch = d2.inch - d1.inch;
            }
return temp;
}

int main()
{

            Distance dis1,dis2,addition,subtraction;
            dis1.input();
            dis2.input();
            addition = add(dis1,dis2);                              //friend function calling
            subtraction = subtract(dis1,dis2);                    //friend function calling
            cout<<endl<<"First Distance object : ";
            dis1.display();
```

```
        cout<<endl<<"Second Distance object : ";
        dis2.display();
        cout<<endl<<"Addition : ";
        addition.display();
        cout<<endl<<"Subtraction : ";
        subtraction.display();
        getch();
        return 0;
}
```

OUTPUT :



```
Enter Feet : 5
Enter Inch : 8
Enter Feet : 8
Enter Inch : 6

First Distance object : Distance is = 5 Feet 8 Inch

Second Distance object : Distance is = 8 Feet 6 Inch

Addition : Distance is = 14 Feet 2 Inch

Subtraction : Distance is = 2 Feet 10 Inch
```

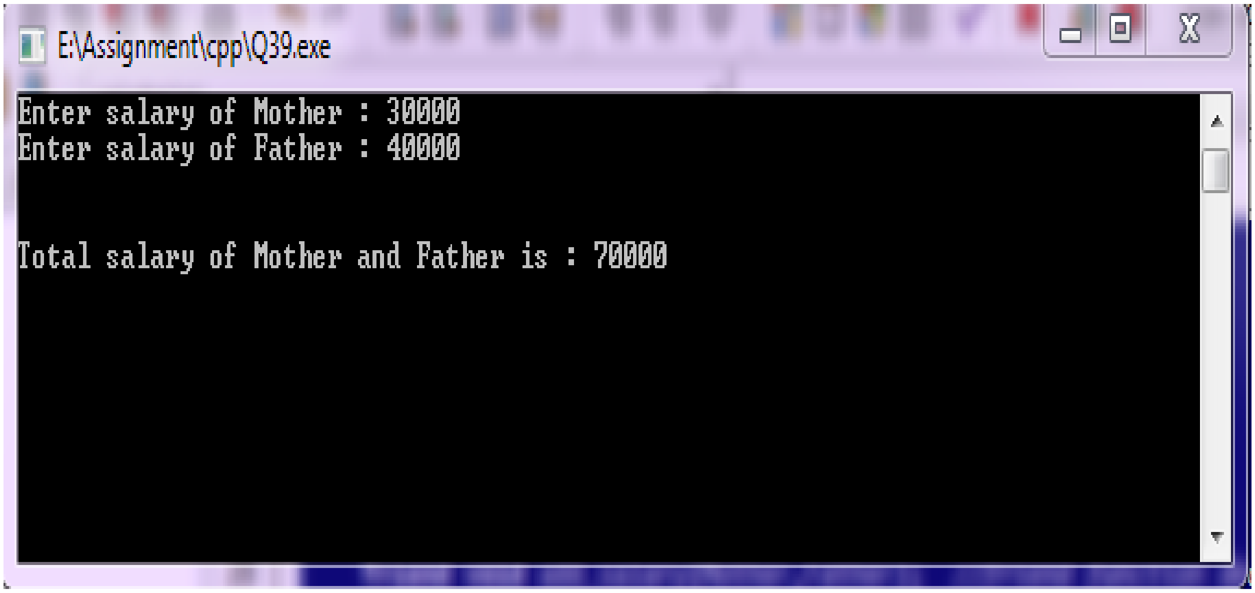
Q39. Write a program to create class Mother having data member to store salary of Mother,create another class Father having data member to store salary of Father. Write a friend function ,which accepts objects of class Mother , and Father and Prints Sum of salary of Mother and Father objects.

PROGRAM :

//Program to add salary of Mother and Father using friend function

```
#include<iostream>
#include<conio.h>
using namespace std;

class Father;                                   //forward declaration of class Father
class Mother
{
        float salary;
public:
        void input()
        {
                cout<<"Enter salary of Mother : ";
                cin>>salary;
        }
        friend void add_salary(Mother,Father);
               //friend function declaration passing Mother and Father type of objcts an argument
};

class Father
{
        float salary;
public:

   void input()
        {
           cout<<"Enter salary of Father : ";
                cin>>salary;
        }
        friend void add_salary(Mother,Father);            //friend function declaration
};

void add_salary(Mother m,Father f)                //friend function definition
{
        float total_sal = m.salary + f.salary ;
        cout<<"Total salary of Mother and Father is : "<<total_sal<<endl;
}
int main()
{
        Mother m1;                      //object created for mother class
        Father f1;                              //object created for father class
        m1.input();
        f1.input();
        cout<<endl<<endl;
        add_salary(m1,f1);
                        //calling friend function and passing objects of mother and father
        getch();
        return 0;
}
```

OUTPUT :



```
Enter salary of Mother : 30000
Enter salary of Father : 40000


Total salary of Mother and Father is : 70000
```

Q40. Write a program to create class having data member to store salary of Mother , create another class Father having data member to store salary of Father. Declare class Father to be friend class of Mother. Write a member function in Father, which accepts object of class Mother and prints Sum of Salary of Mother and Father Objects. Create member function in each class to get input in data member and to display the value of data member.

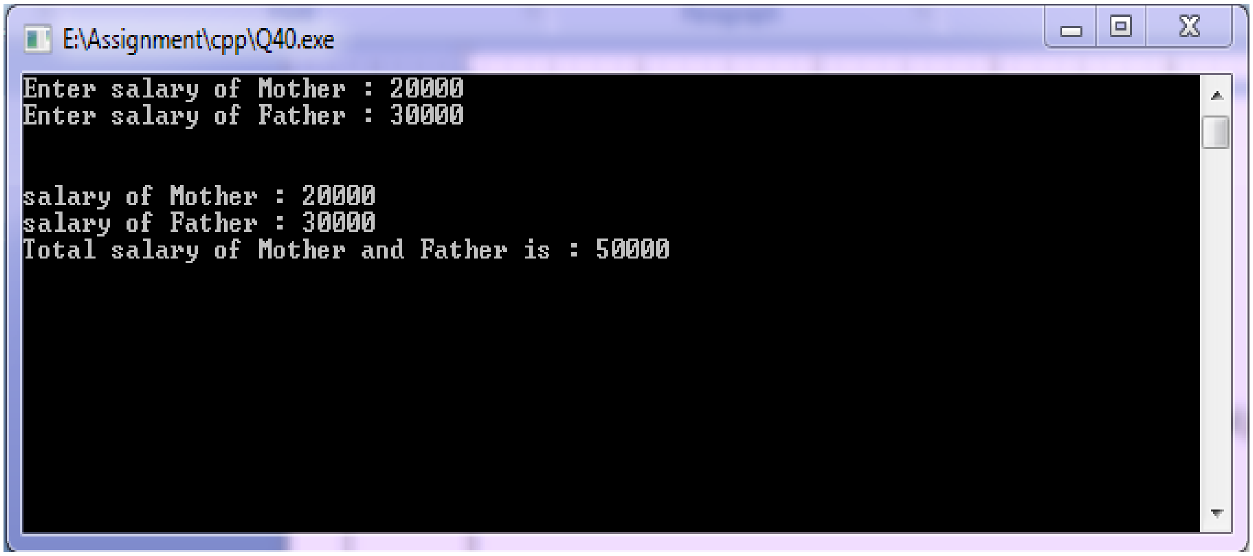
PROGRAM :

//Program to add salary of Mother and Father using friend class

```
#include<iostream>
#include<conio.h>
using namespace std;

class Father;                                   //forward declaration of class Father
class Mother
{
        float salary;
public:
        void input()
        {
                cout<<"Enter salary of Mother : ";
                cin>>salary;
        }
        void display()
        {
                cout<<"salary of Mother : "<<salary<<endl;
        }
  friend class Father;
                                                //friend class declaration
};
class Father
{
        float salary;
public:
   void input()
        {
           cout<<"Enter salary of Father : ";
           cin>>salary;
        }
        void display()
        {
```

```
                    cout<<"salary of Father : "<<salary<<endl;
            }

   void add_salary(Mother);                //Passing Mother type of object an argument of fun.
};

void Father::add_salary(Mother m)               //friend function definition
{
             float total_sal = m.salary + salary ;
            cout<<"Total salary of Mother and Father is : "<<total_sal<<endl;
}
int main()
{
            Mother m1;                  //object created for mother class
            Father f1;                  //object created for father class
            m1.input();
            f1.input();
            cout<<endl<<endl;
            m1.display();
            f1.display();
            f1.add_salary(m1);
                        //calling friend function by class Father object and passing objects of mother
            getch();
            return 0;
}
```

OUTPUT :



```
Enter salary of Mother : 20000
Enter salary of Father : 30000


salary of Mother : 20000
salary of Father : 30000
Total salary of Mother and Father is : 50000
```

Q41. Create a base class shape having two data members with two-member function getdata (pure virtual function) and printarea (not pure virtual function).
Derive classes triangle and rectangle from class shape and redefine member function printarea in both classes triangle and rectangle and test the functioning of classes using pointer to base class objects and normal objects.

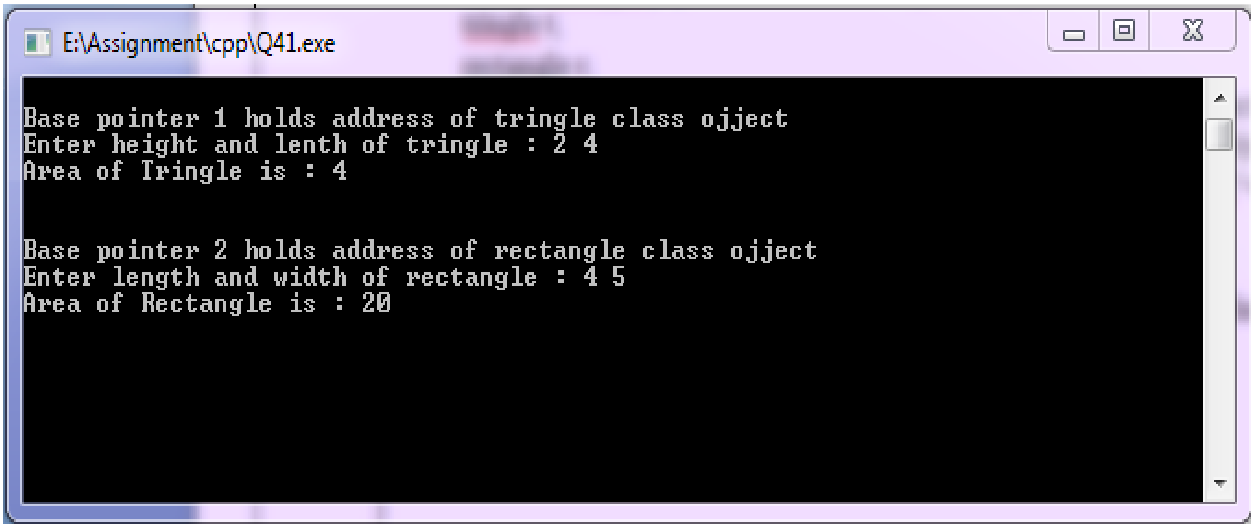
PROGRAM :

//Program to demonstrate pointer to abstract base class

```
#include<iostream>
#include<conio.h>
using namespace std;
class shape                                       //abstract base class
{
  int a;
  float b;

   public:
     virtual void getdata() = 0;                     //pure virtual function definition
             virtual void printarea()                //virtual function definition
             {
                cout<<"Lets print the area you want "<<endl;
             }
};
class tringle:public shape                        //tringle is derived from shape class
{
            float height;
            float length;
            public:
                        void getdata()
                        {
                                    cout<<"Enter height and lenth of tringle : ";
                                    cin>>height>>length;
                        }
                        void printarea()
                        {
                                    cout<<"Area of Tringle is : "<<0.5*height*length<<endl;
                        }
};
class rectangle:public shape               //rectangle is derived from shape class
{
            float width;
```

```
        float length;
        public:
                void getdata()
                {
                        cout<<"Enter length and width of rectangle : ";
                        cin>>length>>width;
                }
                void printarea()
                {
                        cout<<"Area of Rectangle is : "<<length * width<<endl;
                }
};
int main()
{
        shape *base_ptr[2];                     //pointer of base class
        tringle t;
        rectangle r;
        base_ptr[0] = &t;               //holds address of tringle class object
        base_ptr[1] = &r;               //holds address of rectangle class object
        cout<<endl<<"Base pointer 1 holds address of tringle class ojject"<<endl;
        base_ptr[0]->getdata();
        base_ptr[0]->printarea();
        cout<<endl<<endl<<"Base pointer 2 holds address of rectangle class ojject"<<endl;
        base_ptr[1]->getdata();
        base_ptr[1]->printarea();
        getch();
        return 0;
}
```

OUTPUT :



```

Base pointer 1 holds address of tringle class ojject
Enter height and lenth of tringle : 2 4
Area of Tringle is : 4


Base pointer 2 holds address of rectangle class ojject
Enter length and width of rectangle : 4 5
Area of Rectangle is : 20
```