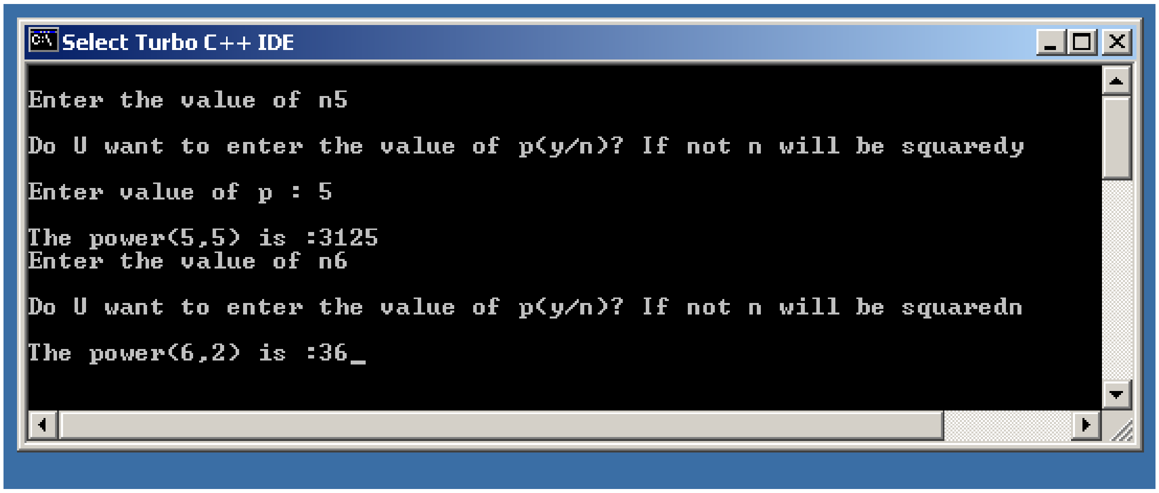
OOPS
Lab Mannual

Program No. 1:- Write a function called power ( ) that takes a double value for n and an int value for p, and returns the result(n^P) as double value. Use a default argument of 2 for p, so that if this argument is omitted, the number will be squared. Write a main ( ) function that gets values from the user to test this function.

Coding:-

```
#include<iostream.h>
#include<conio.h>
double power(double n,int p)
{
        double res=1;
        int i;
        for(i=1;i<=p;i++)
        res=res*n;
        return res;
}
void main()
{
        double n,pow;
        int p;
        char ch;
        clrscr();
        cout<<"\n\nEnter the value of n\t";
        cin>>n;
        cout<<"\nDo U want to enter the value of p(y/n)? If not n will be squared";
        cin>>ch;
        if(ch=='Y'||ch=='y')
        {
                cout<<"\nEnter value of p :\t ";
                cin>>p;
        }
        else p=2;
        pow=power(n,p);
        cout<<"\nThe power("<<n<<","<<p<<") is :"<<pow;
        getch();
}
```

Output of Program:-



```
Enter the value of n5

Do U want to enter the value of p(y/n)? If not n will be squaredy

Enter value of p : 5

The power(5,5) is :3125
Enter the value of n6

Do U want to enter the value of p(y/n)? If not n will be squaredn

The power(6,2) is :36_
```

Program No. 2:- A point on the two dimensional plane can be represented by two numbers: an X coordinate and a Y coordinate. For example, (4,5) represents a point 4 units to the right of the origin along the X axis and 5 units up the Y axis. The sum of two points can be defined as a new point whose X coordinate is the sum of the X coordinates of the points and whose Y coordinate is the sum of their Y coordinates.

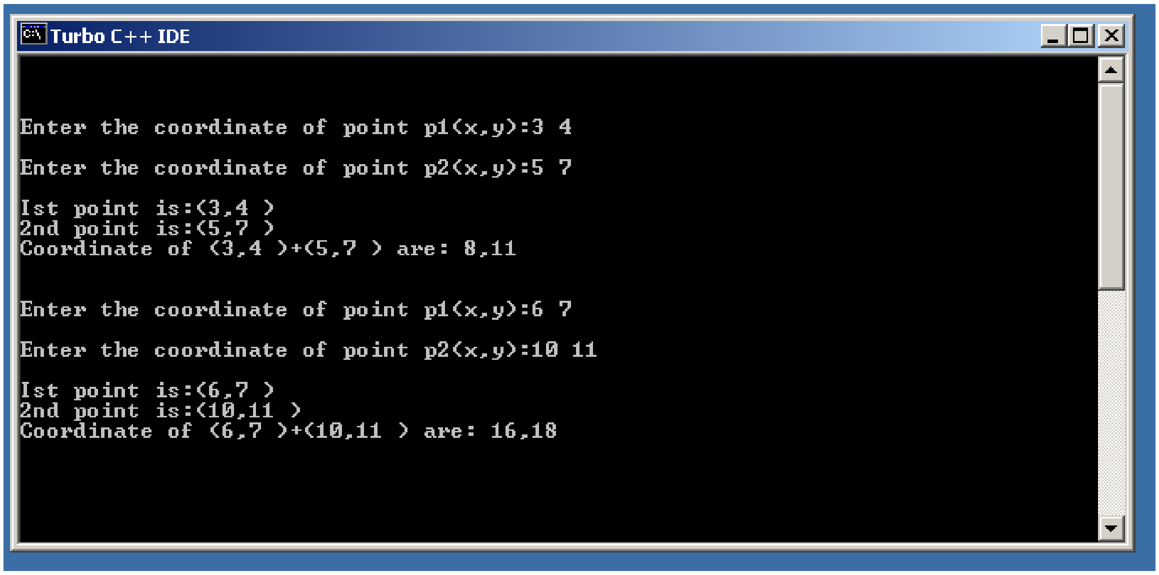
Write a program that uses a structure called point to model a point. Define three points, and have the user input values to two of them. Than set the third point equal to the sum of the other two, and display the value of the new point.

Coding:-

```
#include <iostream.h>
#include <conio.h>
void display(struct point p1);
struct point
{
        int x,y;
};
void sum(struct point p1,struct point p2)
{
        struct point p3;
        p3.x=p1.x+p2.x;
        p3.y=p1.y+p2.y;
        cout<<"\nCoordinate of ";
        display(p1);
        cout<<"+";
        display(p2);
        cout<<" are: "<<p3.x<<","<<p3.y;
}

void display(struct point p1)
{
        cout<<"("<<p1.x<<","<<p1.y<<" )";
}

void main()
{
```

```
                struct point p1,p2,p3;
                clrscr();
                cout<<"\n\n\nEnter the coordinate of point p1(x,y):";
                cin>>p1.x>>p1.y;
                cout<<"\nEnter the coordinate of point p2(x,y):";
                cin>>p2.x>>p2.y;
                cout<<"\nIst point is:";display(p1);
                cout<<"\n2nd point is:";display(p2);
                sum(p1,p2);
                getch();
        }
```

Output of Program:-



```


Enter the coordinate of point p1(x,y):3 4

Enter the coordinate of point p2(x,y):5 7

Ist point is:(3,4 )
2nd point is:(5,7 )
Coordinate of (3,4 )+(5,7 ) are: 8,11


Enter the coordinate of point p1(x,y):6 7

Enter the coordinate of point p2(x,y):10 11

Ist point is:(6,7 )
2nd point is:(10,11 )
Coordinate of (6,7 )+(10,11 ) are: 16,18
```

Program No. 3:-Create the equivalent of a four function calculator. The program should request the user to enter a number, an operator, and another number. carry out the specified arithmetical operation: adding, subtracting, multiplying, or dividing the two numbers. (Using switch statement). Finally it should display the result.

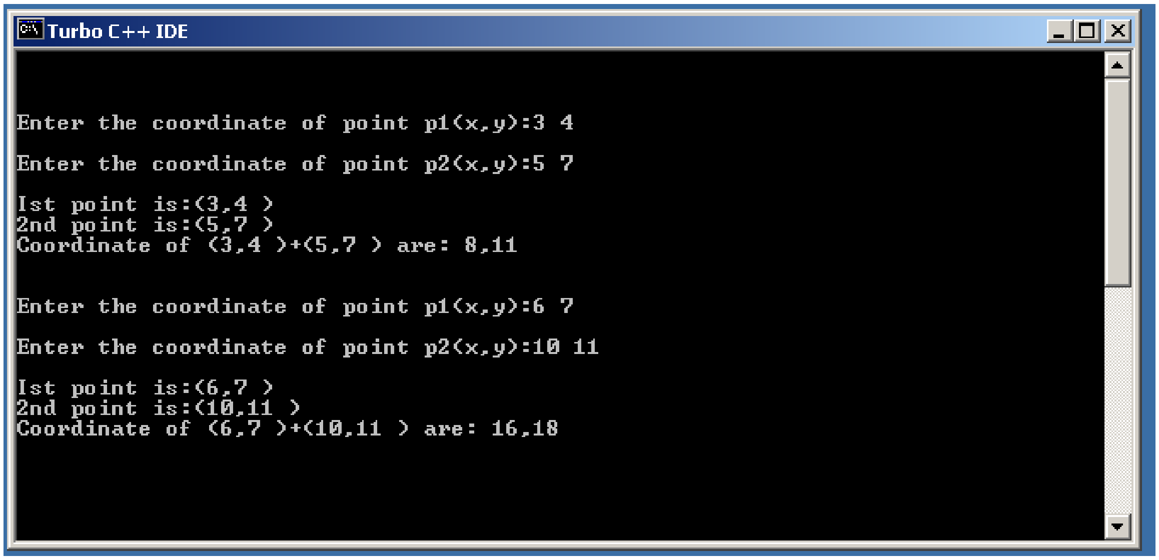
When it finishes the calculation, the program should ask if the user wants to do another calculation. The response can be ‘Y’ or ‘N’

Coding:-

```
#include <iostream.h>
#include<conio.h>
void calculator(int no1,char op,int no2)
{
        float res;
        switch(op)
        {
        case '+':
                res=no1+no2;
                cout<<"Answer = "<<res;
                break;
        case '-':
                res=no1-no2;
                cout<<"Answer = "<<res;
                break;
        case '*':
                res=no1*no2;
                cout<<"Answer = "<<res;
                break;
        case '/':
                res=float(no1)/float(no2);
                cout<<"Answer = "<<res;
                break;
        default :
                cout<<"Invalid operation";
        }
}

void main()
```

```
{
        int n1,n2;
        char op,ch;
        clrscr();
        do
        {
                cout<<"\n\nEnter First number,operator,2nd number:";
                cin>>n1>>op>>n2;
                calculator(n1,op,n2);
                cout<<"\nDo you want to perform another
                calculation(Y/N)?";
                cin>>ch;
        }while(ch=='y'||ch=='Y');
        getch();
}
```

Output of Program:-



```


Enter the coordinate of point p1(x,y):3 4

Enter the coordinate of point p2(x,y):5 7

Ist point is:(3,4 )
2nd point is:(5,7 )
Coordinate of (3,4 )+(5,7 ) are: 8,11


Enter the coordinate of point p1(x,y):6 7

Enter the coordinate of point p2(x,y):10 11

Ist point is:(6,7 )
2nd point is:(10,11 )
Coordinate of (6,7 )+(10,11 ) are: 16,18
```

Program No.4:- A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767) and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers.

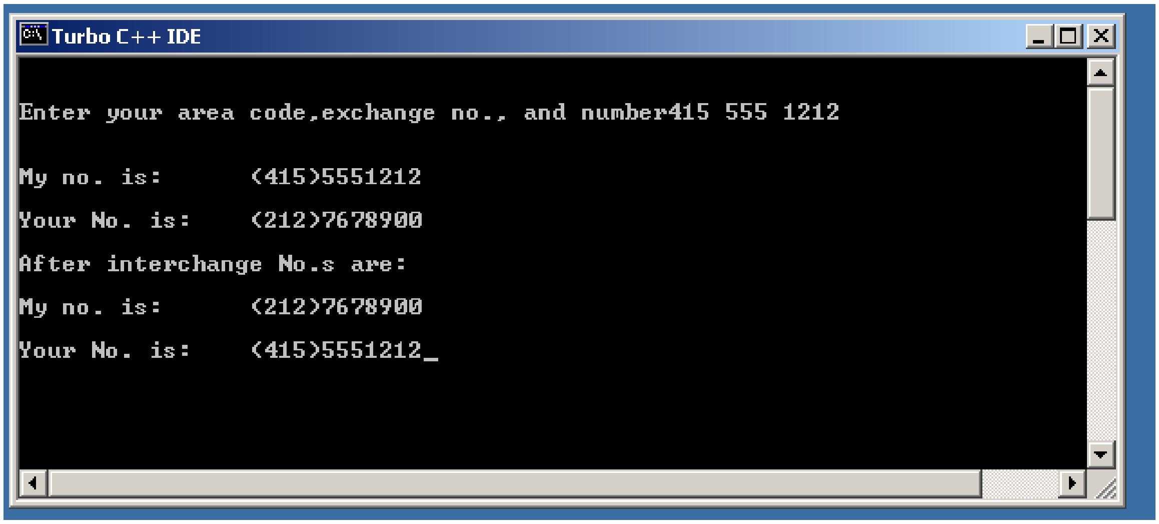
Coding :-

```
#include<iostream.h>
#include<conio.h>

struct ph_no
{
        int area_cod;
        int exchng_no;
```

```
	double num;
};

void display(struct ph_no num)
{
	cout<<"("<<num.area_cod<<")"<< num.exchng_no<<num.num;
}

void swap(struct ph_no *num1,struct ph_no *num2)
{
	struct ph_no num3;
	num3.area_cod=num1->area_cod;
	num3.exchng_no=num1->exchng_no;
	num3.num=num1->num;
	num1->area_cod=num2->area_cod;
	num1->exchng_no=num2->exchng_no;
	num1->num=num2->num;
	num2->area_cod=num3.area_cod;
	num2->exchng_no=num3.exchng_no;
	num2->num=num3.num;
}

void main()
{
	struct ph_no my_no,ur_no;
	clrscr();
	ur_no.area_cod=212;
	ur_no.exchng_no=767;
	ur_no.num=8900;
	cout<<"\n\nEnter your area code,exchange no., and number";
	cin>>my_no.area_cod>>my_no.exchng_no>>my_no.num;
```

```
		cout<<"\n\nMy no. is:\t";
		display(my_no);
		cout<<"\n\nYour No. is:\t";
		display(ur_no);
		swap(&my_no,&ur_no);
		cout<<"\n\nAfter interchange No.s are:";
		cout<<"\n\nMy no. is:\t";
		display(my_no);
		cout<<"\n\nYour No. is:\t";
		display(ur_no);
		getch();
	}
```

Output of Program:-



```
Enter your area code,exchange no., and number415 555 1212

My no. is:      (415)5551212

Your No. is:    (212)7678900

After interchange No.s are:

My no. is:      (212)7678900

Your No. is:    (415)5551212_
```

Program No.5:- Create two classes DM and DB which store the value of distances. DM stores distances in metres and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB.

Use a friend function to carry out the addition operation. The object that stores the results maybe a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or metres and cenitmetres depending on the object on display.

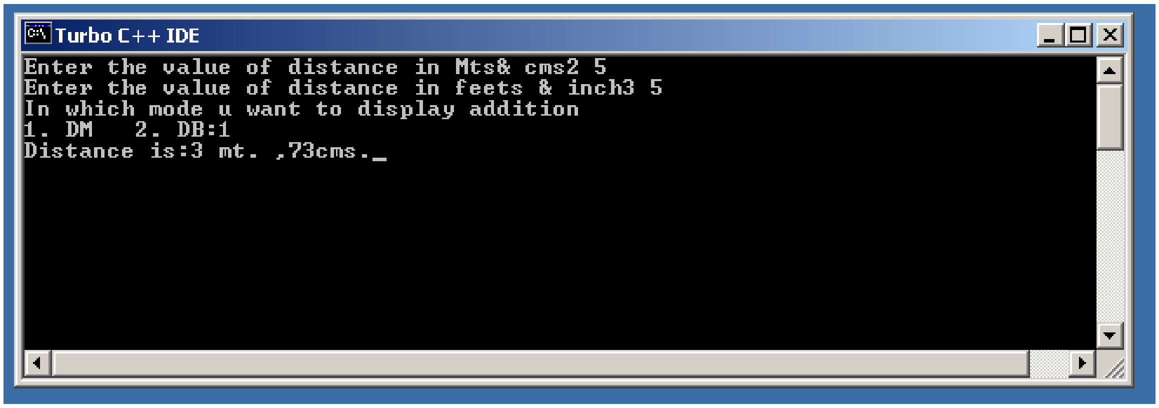
Coding of Program:-

```
#include <iostream.h>
#include<conio.h>
class DB;
class DM
{
        private:int mt,cm;
        public:
        DM(int n1,int n2)
        {
                  mt=n1;
                  cm=n2;
        }
        DM()
        { }
        void getdata()
        {
                  cin>>mt>>cm;
        }
        void display()
        {
                  cout<<"Distance is:"<<mt<<" mt. "<<","<<cm<<"cms.";
        }
        friend DB add(DM ob1,DB ob2);
        friend DM DB2DM(DB obj);
};

class DB
{
        private:int ft,inch;
        public:
        DB(int n1,int n2)
        {
                  ft=n1;
                  inch=n2;
        }
        DB()
        { }
        void getdata()
```

```
    {
        cin>>ft>>inch;
    }
    void display()
    {
        cout<<"Distance is:"<<ft<<" feet "<<","<<inch<<"inch";
    }

    friend DM DB2DM(DB obj);
    friend DB add(DM ob1,DB ob2);
} ;


DB add(DM ob1,DB ob2)
{
    DB db;
    int dm_inch=(ob1.mt*100+ob1.cm)/2.54;
    int t_inch=(dm_inch+ob2.inch)/12;
    db.ft=ob2.ft+t_inch;
    db.inch=(t_inch+ob2.inch)%12;
    return db;
}
DM DB2DM(DB obj1)
{
    DM obj;
    int t_cms=(obj1.ft*12+obj1.inch)*2.54;
    obj.mt=t_cms/100;
    obj.cm=t_cms%100;
    return obj;
}
```

```
void main()
{
	DB d1,d3;
	int mode;
	DM d2;
	cout<<"Enter the value of distance in Mts& cms";
	d1.getdata();
	cout<<"Enter the value of distance in feets & inch";
	d2.getdata();
	cout<<"In which mode u want to display addition\n1. DM\t2. DB:";
	cin>>mode;
	d3=add(d2,d1);
	if (mode==2)
	{
		d3.display();
	}
	else
	{
		DM  d4=DB2DM(d3);
		d4.display();
	}
	getch();
}
```

Output of Program:-



```
Enter the value of distance in Mts& cms2 5
Enter the value of distance in feets & inch3 5
In which mode u want to display addition
1. DM   2. DB:1
Distance is:3 mt. ,73cms.
```

Program No. 6:- Create a class rational which represents a numerical value by two double values- NUMERATOR & DENOMINATOR. Include the following public member functions:

-constructor with no arguments (default).

-constructor with two arguments.

-void reduce( ) that reduces the rational number by eliminating the highest common factor .

-Overload + operator to add two rational number.

- Overload >> operator to enable input through cin.

-Overload << operator to enable output through cout

Write a main ( ) to test all the functions in the class.

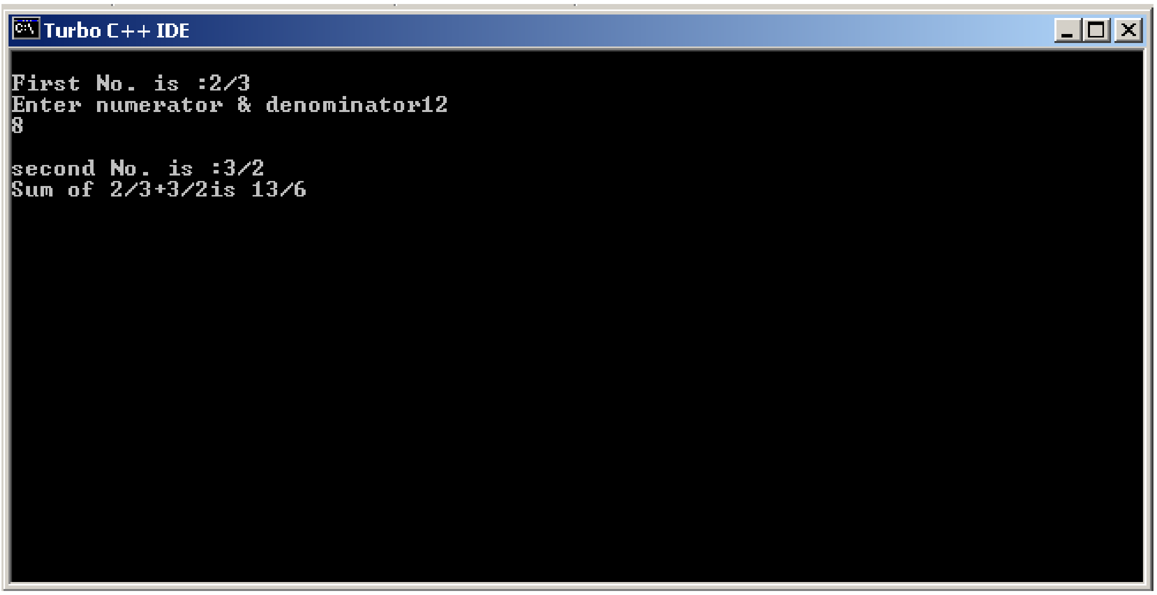
Coding of Program:-

```
#include<iostream.h>
#include<conio.h>
```

```
class Rational
{
private:
        int numerator,denominator;
public:
        Rational()
        {
                numerator=0;
                denominator=0;
        }

        Rational(int a,int b)
        {
                numerator=a;
                denominator=b;
        }

        void reduced()
        {
                int p;
                if(numerator>denominator)
                p=denominator;
                else p=numerator;
                for(int i=2;i<=p;i++)
                while(numerator%i==0 && denominator%i==0)
                {
                        numerator=numerator/i;
                        denominator=denominator/i;
                }
        }

        Rational operator +(Rational b)
        {
                Rational c;

        c.numerator=numerator*b.denominator+b.numerator*denominator
;
                c.denominator=denominator*b.denominator;
                c.reduced();
                return c;
        }
        friend ostream& operator << (ostream& out,Rational &r);
        friend istream& operator >> (istream& in, Rational &r);
};
        ostream& operator << (ostream& out,Rational& r)
```

```
    {      r.reduced();
           out<<r.numerator<<"/"<<r.denominator;
           return out;
    }

    istream& operator >>(istream& in,Rational& r)
    {

           cout<<"\nEnter numerator & denominator";
           in>>r.numerator>>r.denominator;
           return in;
    }

void main()
{
    Rational a(10,15),b,c;
    clrscr();
    cout<<"\nFirst No. is :"<<a;
    cin>>b;
    cout<<"\nsecond No. is :"<<b;
    c=a+b;
    cout<<"\nSum of "<<a<<"+"<<b<<"is "<<c;
    getch();
}
```

Output of Program:-



```

First No. is :2/3
Enter numerator & denominator12
8

second No. is :3/2
Sum of 2/3+3/2is 13/6
```

Program No. 7:- Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data ( ) to initialize base class data members and another member function display_area ( ) to compute and display the area of figures. Make display_area ( ) as a virtual function and redefine this function in the derived classes to suit their requirements.

Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display the area. Remember the two values given as input will be treated as lengths of two sides in the case of rectangles and as base and height in the case of triangles and used as follows:

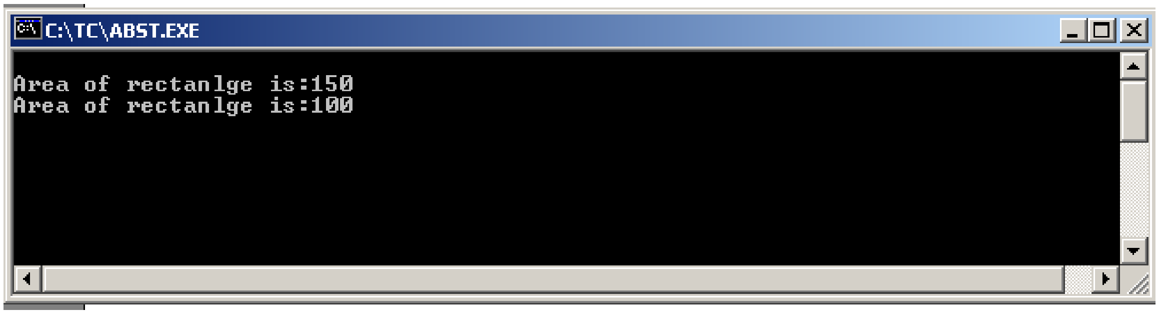
Area of rectangle	= x * y
Area of triangle	= ½ * x * y

Coding of Programm:-

```
#include<iostream.h>

#include<conio.h>

class shape

{

        protected:

                double dim1,dim2,area;

        public:

                void get_data(double a,double b)

                {

                        dim1=a;

                        dim2=b;

                }

                virtual void display_area()=0;

} ;

class rectangle : public shape

{

public:

                void display_area()     //overridding of base class method

                {

                        area=dim1*dim2;

                        cout<<"\nArea of rectanlge is:"<<area;

                }

};
```

```
class triangle : public shape
{
public:
        void display_area()    //overridding of base class  method
        {
                area=dim1*dim2/2;
                cout<<"\nArea of rectanlge is:"<<area;
        }
};

void main()
{
        //shape s1;       //compile time error as shape is an abstract class
        shape *sptr;
        clrscr();
        rectangle rec1;
        triangle tri1;
        rec1.get_data(10,15);
        tri1.get_data(10,20);
        sptr=&rec1; //assign address of rectangle class object
        sptr->display_area(); // call display_area of rectangle class
        sptr=&tri1; //assign address of an object of triangle derived class
        sptr->display_area();  //call display_area of triangle class
        getch();
}
```

Output of Programm:-



```
Area of rectanlge is:150
Area of rectanlge is:100
```

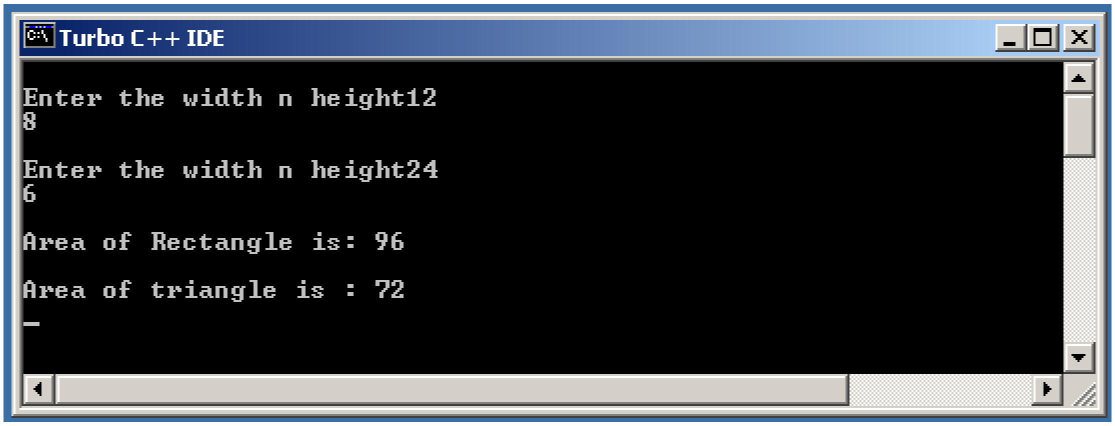
Program No. 8:- Write a Program to implement multiple inheritance.

Coding of Program:-

```
#include<conio.h>
#include <iostream.h>
class CPolygon
{
  protected:
              int width, height;
  public:
             void set_values (int a, int b)
            {
                     width=a; height=b;
             }
            void getdata()
            {
                     cout<<"\nEnter the width n height";
                     cin>>width>>height;
             }
  };

class Coutput
{
```

```
            public:
            void output (int i);
};

void COutput::output (int i)
{
            cout << i << endl;
}

class CRectangle: public CPolygon, public COutput
{
public:
            int area ()
            {
            return (width * height);
             }
  };

class CTriangle: public CPolygon, public COutput
{
  public:
           int area ()
          {
           return (width * height / 2);
           }
};

int main ()
{
CRectangle rect
CTriangle trgl;
clrscr();
rect.getdata();
trgl.getdata();
cout<<"\nArea of Rectangle is: ";
rect.output (rect.area());
cout<<"\nArea of triangle is : ";
trgl.output (trgl.area());
getch();
return 0;
}
```

Output of Program:-



```
Enter the width n height12
8

Enter the width n height24
6

Area of Rectangle is: 96

Area of triangle is : 72
_
```

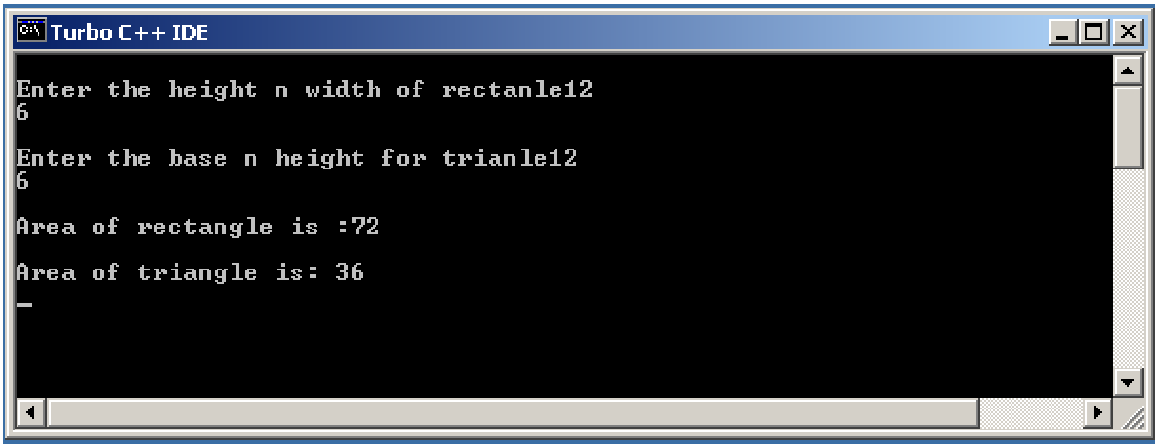
Program No. 9:- Write a program to implement abstract class.

Coding of Program:-

```
#include <iostream.h>
#include<conio.h>

class Shape
{
protected:
                int width, height;
        public:
                void set_values (int a, int b)
                {
                  width=a; height=b;
                 }
                virtual int area (void) =0;
                virtual void getdata()=0;
                void printarea (void)
                {
                  cout << this->area() << endl;
                }
```

```
  };

class CRectangle: public Shape
{
          public:
                    int area (void)
                    {
                      cout<<"\nArea of rectangle is :";
                      return (width * height);
                    }
                    void getdata()
                   {
                     cout<<"\nEnter the height n width of rectanle";
                     cin>>width>>height;
                   }
  };

class CTriangle: public Shape
{
          public:
                    int area (void)
                    {
                      cout<<"\nArea of triangle is: ";
                      return (width * height / 2);
                   }
                  void getdata()
                  {
                    cout<<"\nEnter the base n height for trianle";
                    cin>>width>>height;
                  }
  };

int main ()
{
          CRectangle rect;
          CTriangle trgl;
           clrscr();
          Shape *s1 = &rect;
          Shape * s2 = &trgl;
          s1->getdata();
          s2->getdata();
          s1->printarea();
          s2->printarea();
          getch();
          return 0;
}
```

Output of Program:-



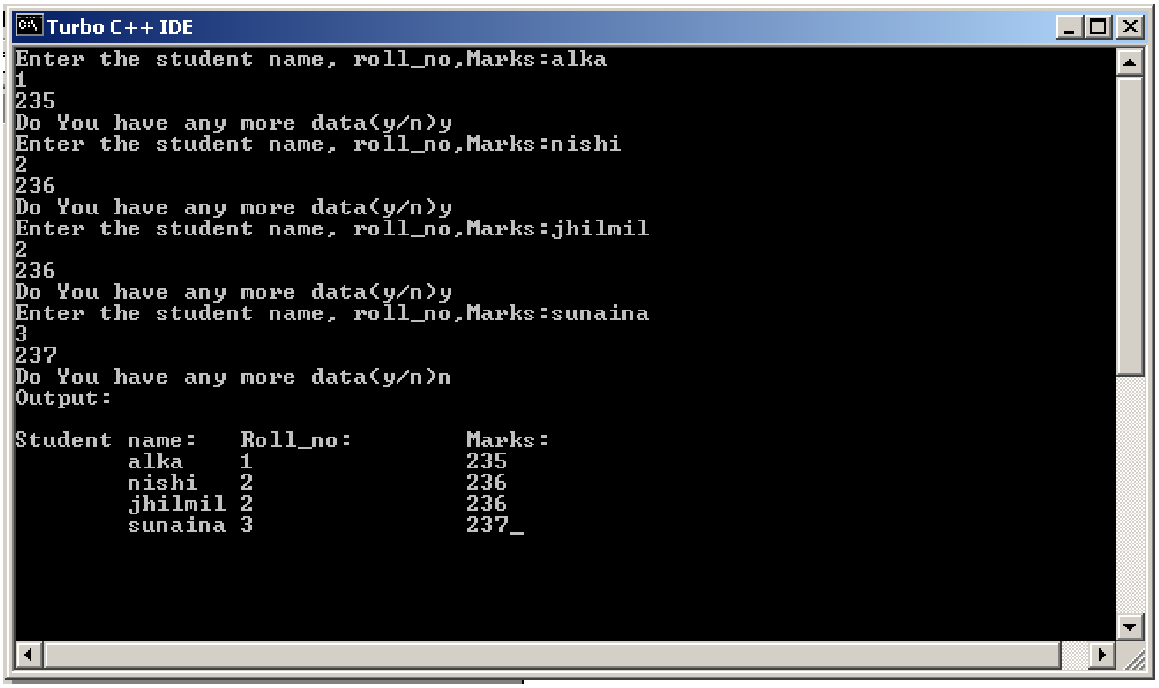
```
Enter the height n width of rectanle12
6

Enter the base n height for trianle12
6

Area of rectangle is :72

Area of triangle is: 36
_
```

Program No. 10:- W.A.P. that creates a binary file by reading the data for the students from the terminal. The data of each student consist of roll no., name (a string of 30 or lesser no. of characters) and marks.

Coding of Program:-

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>

struct student
{
int roll_no;
char nm[30];
float marks;
};

void main()
{
struct student st;
char ch;
fstream outfile;
clrscr();
outfile.open("stud.txt",ios::in|ios::out);
do
{
cout<<"Enter the student name, roll_no,Marks:";
cin>>st.nm>>st.roll_no>>st.marks;
outfile.write((char *) &st,sizeof(st));
cout<<"Do You have any more data(y/n)";
cin>>ch;
}while(ch=='y'||ch=='Y');
outfile.seekg(0);
cout<<"Output:\n";
cout<<"\nStudent name:\tRoll_no:\tMarks:";
while(outfile.read((char *) &st,sizeof(st)))
{

cout<<"\n\t"<<st.nm<<"\t"<<st.roll_no<<"\t\t"<<st.marks;
}
outfile.close();
getch();
}
```

Output of Program:-



```
Enter the student name, roll_no,Marks:alka
1
235
Do You have any more data(y/n)y
Enter the student name, roll_no,Marks:nishi
2
236
Do You have any more data(y/n)y
Enter the student name, roll_no,Marks:jhilmil
2
236
Do You have any more data(y/n)y
Enter the student name, roll_no,Marks:sunaina
3
237
Do You have any more data(y/n)n
Output:

Student name:   Roll_no:        Marks:
        alka    1               235
        nishi   2               236
        jhilmil 2               236
        sunaina 3               237
```

Program No. 11:- A hospital wants to create a database regarding its indoor patients. The information to store include

a) Name of the patient
b) Date of admission
c) Disease
d) Date of discharge

Create a structure to store the date (year, month and date as its members). Create a base class to store the above information. The member function should include functions to enter information and display a list of all the patients in the database. Create a derived class to store the age of the patients. List the information about all the to store the age of the patients. List the information about all the pediatric patients (less than twelve years in age).

Coding of Program:-

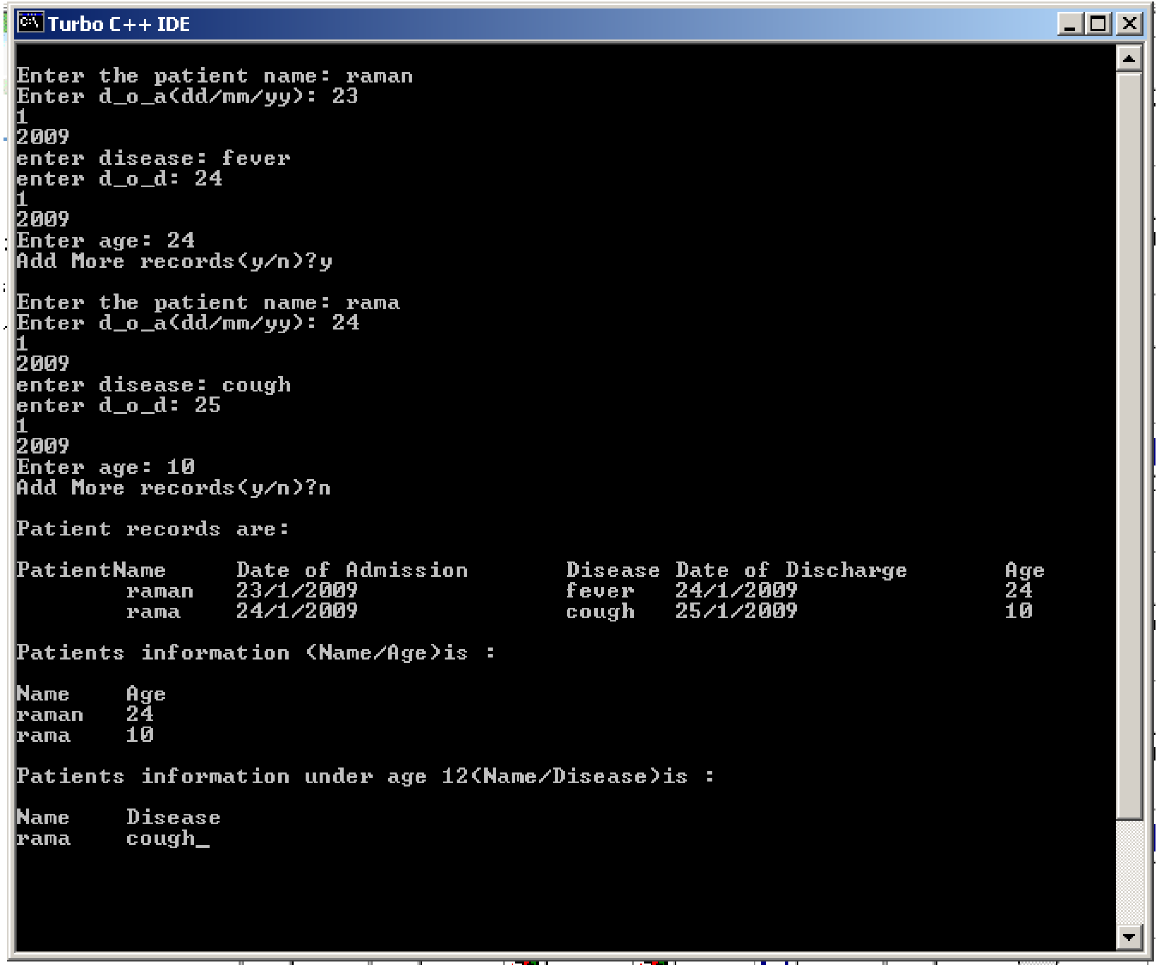
```
#include<conio.h>
#include<iostream.h>
#include<fstream.h>
#include<stdio.h>

struct date
{
        int dd;
        int mm;
        int yy;

};
```

```
class patient
{
protected:
        char nm[30];
        struct date d_o_a,d_o_d;
        char dis[35];
        public:
        void getdata()
        {
                cout<<"\nEnter the patient name: ";
                cin>>nm;
                cout<<"Enter d_o_a(dd/mm/yy): ";
                cin>>d_o_a.dd>>d_o_a.mm>>d_o_a.yy;
                cout<<"enter disease: ";
                cin>>dis;
                cout<<"enter d_o_d: ";
                cin>>d_o_d.dd>>d_o_d.mm>>d_o_d.yy;
        }
        void putdata()
        {

                cout<<"\n\t"<<nm<<"\t"<<d_o_a.dd<<"/"<<d_o_a.mm<<"
                /"<<d_o_a.yy<<"\t\t"<<dis<<"\t"<<d_o_d.dd<<"/"<<d_o_
                d.mm<<"/"<<d_o_d.yy;
        }
};

class pat_drv : protected patient
{
        int age;
        public:
        void getdata()
        {
                patient::getdata();
                cout<<"Enter age: ";
                cin>>age;
        }

        void putdata()
        {
                patient::putdata();
                cout<<"\t\t"<<age;
        }
        void list_info() ;
        void list_under12();
```

```
};
void pat_drv::list_info()
{
    ifstream infile;
    infile.open("pat2.txt");
    infile.seekg(0);
    cout<<"\n\nName\tAge";
    while(infile.read((char *) this,sizeof(*this)))
      cout<<"\n"<<nm<<"\t"<<age;
    infile.close();
}
void pat_drv::list_under12()
{
    ifstream infile;
    infile.open("pat2.txt");
    cout<<"\n\nName\tDisease";
    while(infile.read((char *) this,sizeof(*this)))
        if(this->age<12)
            cout<<"\n"<<nm<<"\t"<<dis;
    infile.close();
}

void main()
{
    fstream outfile;
    outfile.open("pat2.txt",ios::in|ios::out);
    char ch;
    pat_drv p1;
    clrscr();
    do
    {
        p1.getdata();
        outfile.write((char *) & p1,sizeof(p1));
        cout<<"Add More records(y/n)?";
        cin>>ch;
    }while(ch=='y'||ch=='Y');
    fflush(stdin);
    outfile.seekg(0);
    cout<<"\nPatient records are:\n\nPatientName\tDate of
Admission\tDisease\tDate of Discharge\tAge";
    while(outfile.read((char *)&p1,sizeof(p1)))
    {
    p1.putdata();
    }
    outfile.close();
    cout<<"\n\nPatients information (Name/Age)is :";
```

```
            p1.list_info();
            cout<<"\n\nPatients information under age 12(Name/Disease)is :";
            p1.list_under12();
            getch();
    }
```

Output of Program:-



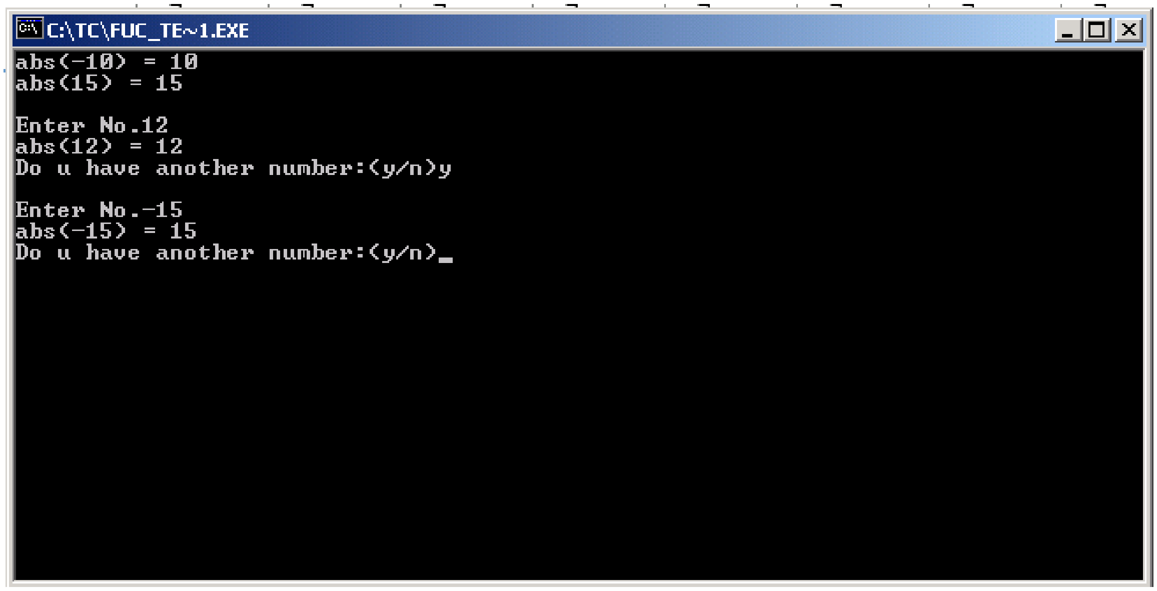
```
Enter the patient name: raman
Enter d_o_a(dd/mm/yy): 23
1
2009
enter disease: fever
enter d_o_d: 24
1
2009
Enter age: 24
Add More records(y/n)?y

Enter the patient name: rama
Enter d_o_a(dd/mm/yy): 24
1
2009
enter disease: cough
enter d_o_d: 25
1
2009
Enter age: 10
Add More records(y/n)?n

Patient records are:

PatientName     Date of Admission       Disease Date of Discharge       Age
        raman   23/1/2009               fever   24/1/2009               24
        rama    24/1/2009               cough   25/1/2009               10

Patients information (Name/Age)is :

Name    Age
raman   24
rama    10

Patients information under age 12(Name/Disease)is :

Name    Disease
rama    cough_
```

Program No. 12:- W.A.P. to implement abs( ) function using function template so that it will work with any basic numerical type.
Abs( ) function return the absolute value of a number.

Coding of Program:-

```
#include <iostream.h>
#include<conio.h>

//abs returns the absolute value of a number.
template <class T>
T abs(T a)
{
        return a >=0 ? a : -a ;
}


void main()
{
        clrscr();
        cout << "abs(-10) = " << abs(-10) << endl ;
        cout << "abs(15) = " << abs(15) << endl ;
        int i;
        char ch;
        do
        {
        cout<<"\nEnter No.";
        cin>>i;
        cout << "abs("<<i<<") = " << abs(i) << endl ;
        cout<<"Do u have another number:(y/n)";
        cin>>ch;
        }while(ch=='y'||ch=='Y');
        getch();
}
```

Output of Program:-



```
abs(-10) = 10
abs(15) = 15

Enter No.12
abs(12) = 12
Do u have another number:(y/n)y

Enter No.-15
abs(-15) = 15
Do u have another number:(y/n)_
```

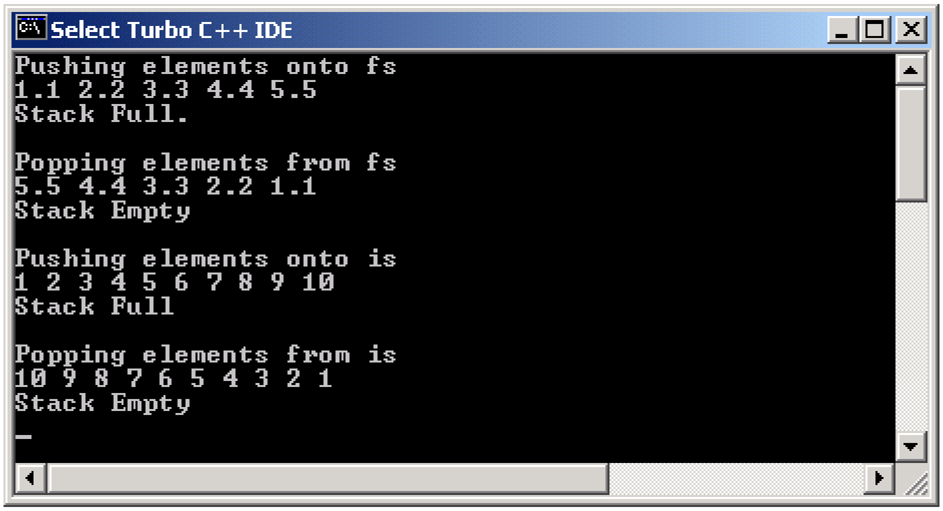
Program No. 13:- W.A.P. to implement Stack and its operation using Template class so that it will work with stack of int, float, char types.

Coding of Program:-

```
#include<iostream.h>
#include<conio.h>
template <class T>
class Stack
{
public:
        Stack(int = 10) ;
        ~Stack() { delete [] stackPtr ; }
        int push(const T&);
        int pop(T&) ;  // pop an element off the stack
        int isEmpty()const { return top == -1 ; }
        int isFull() const { return top == size - 1 ; }
private:
        int size ;  // Number of elements on Stack
        int top ;
        T* stackPtr ;
} ;

//constructor with the default size 10
template <class T>
Stack<T>::Stack(int s)
{
        size = s > 0 && s < 1000 ? s : 10 ;
        top = -1 ;  // initialize stack
        stackPtr = new T[size] ;
}
 // push an element onto the Stack
template <class T>
int Stack<T>::push(const T& item)
{
        if (!isFull())
        {
                stackPtr[++top] = item ;
                return 1 ;  // push successful
        }
        return 0 ;  // push unsuccessful
}

// pop an element off the Stack
template <class T>
int Stack<T>::pop(T& popValue)
{
```

```
            if (!isEmpty())
            {
                        popValue = stackPtr[top--] ;
                        return 1 ;  // pop successful
            }
            return 0 ;  // pop unsuccessful
}

void main()
{
            typedef Stack<float> FloatStack ;
            typedef Stack<int> IntStack ;

            FloatStack fs(5) ;
            clrscr();
            float f = 1.1 ;
            cout << "Pushing elements onto fs" << endl ;
            while (fs.push(f))
            {
                        cout << f << ' ' ;
                        f += 1.1 ;
            }
            cout << endl << "Stack Full." << endl
            << endl << "Popping elements from fs" << endl ;
            while (fs.pop(f))
                        cout << f << ' ' ;
            cout << endl << "Stack Empty" << endl ;
            cout << endl ;

            IntStack is ;
            int i = 1.1 ;
            cout << "Pushing elements onto is" << endl ;
            while (is.push(i))
            {
                        cout << i << ' ' ;
                        i += 1 ;
            }
            cout << endl << "Stack Full" << endl
            << endl << "Popping elements from is" << endl ;
            while (is.pop(i))
                                    cout << i << ' ' ;
            cout << endl << "Stack Empty" << endl ;
            getch();
}
```

Output of Program:-



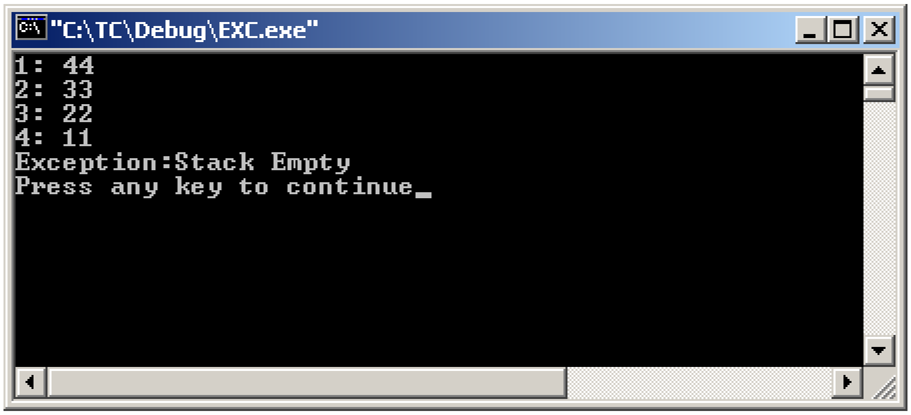
```
Pushing elements onto fs
1.1 2.2 3.3 4.4 5.5
Stack Full.

Popping elements from fs
5.5 4.4 3.3 2.2 1.1
Stack Empty

Pushing elements onto is
1 2 3 4 5 6 7 8 9 10
Stack Full

Popping elements from is
10 9 8 7 6 5 4 3 2 1
Stack Empty
_
```

Program No. 14:- W.A.P. using exception handling to throw separate exception for attempting to push data on a full stack and attempting to pop data from an empty stack

Coding of Program:-

```
#include<iostream.h>
#include<conio.h>

const int MAX=4;

class stack
{
private:
        int st[MAX];
        int top;

public:
        class Full{};
        class Empty{};

        stack ()
        {
        top=-1;
        }

        void push(int var)
        {
        if(top>=MAX-1)
        throw Full();
        st[++top]=var;
        }

        int pop()
        {
        if(top<0)
        throw Empty();
        return st[top--];
        }
};

void main()
{
stack s1;
clrscr();
try
{
s1.push(11);
s1.push(22);
s1.push(33);
s1.push(44);
cout<<"1: "<<s1.pop()<<endl;
cout<<"2: "<<s1.pop()<<endl;
cout<<"3: "<<s1.pop()<<endl;
cout<<"4: "<<s1.pop()<<endl;
cout<<"5: "<<s1.pop()<<endl;
}
catch(stack:: Full)
{
cout<<"Exception:Stack Full"<<endl;
}
catch(stack::Empty)
{
cout<<"Exception:Stack Empty"<<endl;
}
getch();
}
```

Output of Program:-



```
1: 44
2: 33
3: 22
4: 11
Exception:Stack Empty
Press any key to continue_
```