GREEN WOODS ARTS AND SCIENCE COLLEGE

(Affiliated to Kannur University)

AK ROAD, BEKAL, PALAKUNU, KASARAGOD-671318

BACHELOR OF COMPUTER APPLICATIONS

NAME : …………………………………………………………………………………….

REG.NO: …………………………………………………………………………………….

SUBJECT: ……………………………………………………………………………………..

SUBJECT CODE: ……………………………………………………………………………..

GREENWOODS ARTS AND SCIENCE COLLEGE

(Affiliated to KANNUR UNIVERSITY)

A.K Road, P.O Bekal, Kasaragod-671318

CERTIFICATE

Certified that this is bonafide record of the work done by Ms……………………………………………… of ………… semester BCA of GreenWoods Arts and Science College during the academic year ……………..

Lecturer in charge

Head of the dept.

Valued Reg.No………………………..of ……………………

Examiner:

Place:

Date:

INDEX

NO	TITLE	PAGE	DATE

PROGRAM 1

To implement output and reference parameters

A class that declares overloaded functions (function name Add) to find the sum of
integers and concatenate strings with

• variable number of integer arguments or string arguments
• an out parameter to hold the ouput of sum or concatenated string
• a reference parameter with "NULL" message which will be changed to

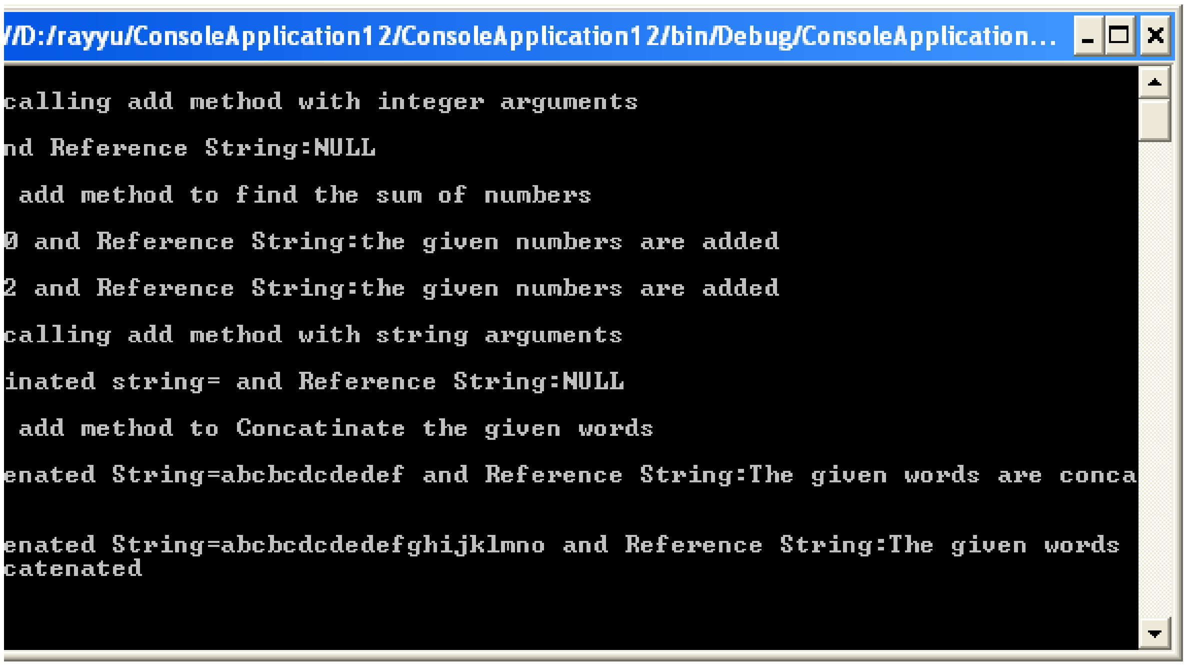
"UPDATED" after calling the function
• after calling, if the message is "UPDATED" then the results are displayed otherwise display "Unable to process.........."

SOURCE CODE

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication12
{
   public class sample
   {
       public void add(out int res, ref string op, params int[] nums)
   {
```

```
            res = 0;
            foreach (int x in nums)
                res = res + x;
            op = "the given numbers are added";
        }
        public void add(out string res, ref string op, params string[] words)
        {
            string str1 = "";
            foreach (string word in words)
            {
                string str = str1 + word;
                str1 = str;
            }
            res = str1;
            op = "The given words are concatenated";
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            sample s = new sample();
            int num_res = 0;
            string string_res, output = "NULL";
            Console.WriteLine("\nBefore calling add method with integer
arguments\n\nSum={0} and Reference String:{1}", num_res, output);
            Console.WriteLine("\nCalling add method to find the sum of numbers");
            s.add(out num_res, ref output, 10, 20, 30, 40, 50, 60);
            Console.WriteLine("\n Sum={0} and Reference String:{1}", num_res, output);
            s.add(out num_res, ref output, 10, 20, 30, 40, 50, 60, 22, 56, 32, 12);
            Console.WriteLine("\n Sum={0} and Reference String:{1}", num_res, output);
            string_res = "";
            output = "NULL";


Console.WriteLine("\nBefore calling add method with string arguments\n\n Concatinated
string={0} and Reference String:{1}", string_res, output);
            Console.WriteLine("\nCalling add method to Concatinate the given words");
            s.add(out string_res, ref output, "abc", "bcd", "cde", "def");
Console.WriteLine("\n Concatenated String={0} and Reference String:{1}", string_res,
output);
            s.add(out string_res, ref output, "abc", "bcd", "cde", "def", "ghi", "jkl", "mno");
            Console.WriteLine("\n Concatenated String={0} and Reference String:{1}",
string_res, output);
            Console.ReadLine();
        }
    }
}
```

OUTPUT


/D:/rayyu/ConsoleApplication12/ConsoleApplication12/bin/Debug/ConsoleApplication...

```
calling add method with integer arguments
nd Reference String:NULL
 add method to find the sum of numbers
0 and Reference String:the given numbers are added
2 and Reference String:the given numbers are added
calling add method with string arguments
inated string= and Reference String:NULL
 add method to Concatinate the given words
enated String=abcbcdcdedef and Reference String:The given words are conca

enated String=abcbcdcdedefghijklmno and Reference String:The given words
catenated
```

<u>PROGRAM 2</u>

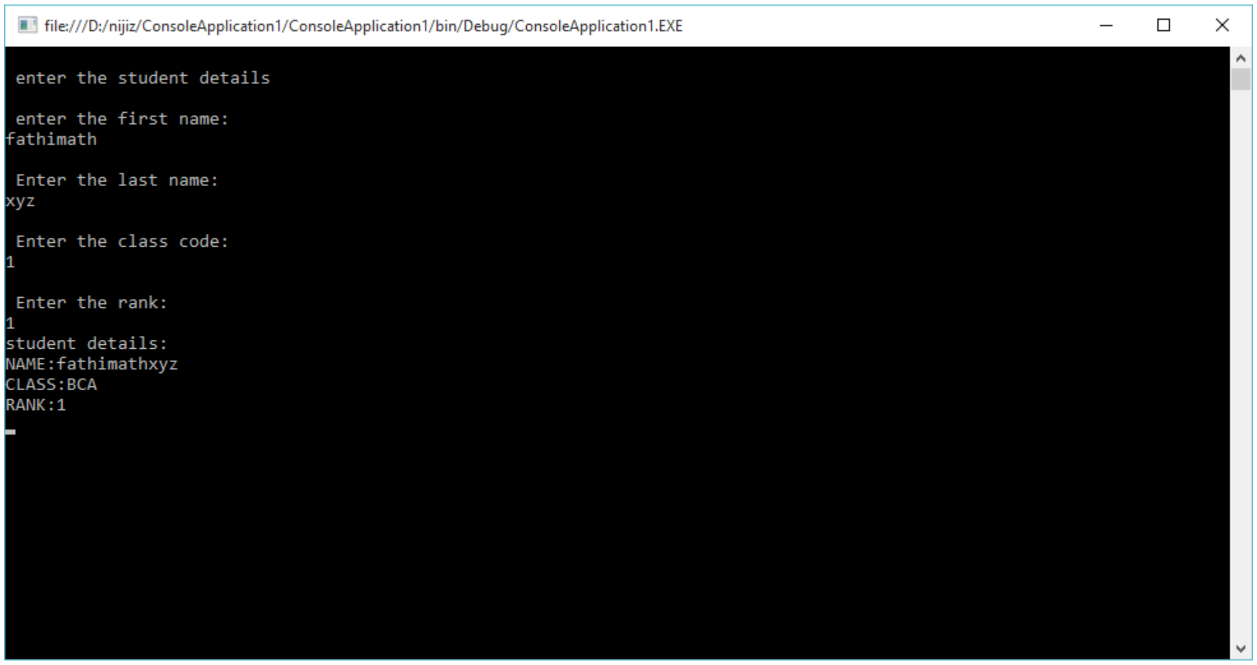
<u>To implement indexers and properties</u>

a) create class with student details first name, last name class, and rank as private
members and declare properties Name, Class and Rank where Name should refer the
firstname and last name (Name = firstname + lastname) concated, Class should give the
name of the class code (like if class=01 then BCA, else if class=02 then BBM etc.,)
given in 'class' member

<u>SOURCE CODE</u>

```
using System.Text;

namespace ConsoleApplication1
{
  public class student
  {
      String firstname,lastname;
      int _class;
      int rank;
      public string Name
      {
          get  {return firstname+lastname;}
      }
      public string Class
      {
          get
          {
          if(_class==1)
              return "BCA";
          else if(_class==2)
              return "BBA";
          else if(_class==3)
              return "Bcom";
          else
          return "INVALID CODE";
      }
  }
    public  int Rank
    {
    set{rank=value;}
    get{return rank;}
}
    public student(string fn,string ln,int cl,int rnk)
{
    firstname=fn;
lastname=ln;
    _class=cl;
rank=rnk;
}
}

    class Program
    {
        public static void Main()
        {
            student stud;
            Console.WriteLine("\n enter the student details");
            Console.WriteLine("\n enter the first name:");
            string fn=Console.ReadLine();
            Console.WriteLine("\n Enter the last name:");
            string ln=Console.ReadLine();
            Console.WriteLine("\n Enter the class code:");
            int cl=Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("\n Enter the rank:");
            int rnk=Convert.ToInt32(Console.ReadLine());
            stud= new student(fn,ln,cl,rnk);
            //using properties to access private values
            Console.WriteLine("student details:");
            Console.WriteLine("NAME:{0}",stud.Name);
            Console.WriteLine("CLASS:{0}",stud.Class);
            Console.WriteLine("RANK:{0}",stud.Rank);

           Console.ReadKey();
        }
      }
  }
```

OUTPUT



file:///D:/nijiz/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.EXE

```
 enter the student details

 enter the first name:
fathimath

 Enter the last name:
xyz

 Enter the class code:
1

 Enter the rank:
1
student details:
NAME:fathimathxyz
CLASS:BCA
RANK:1
```

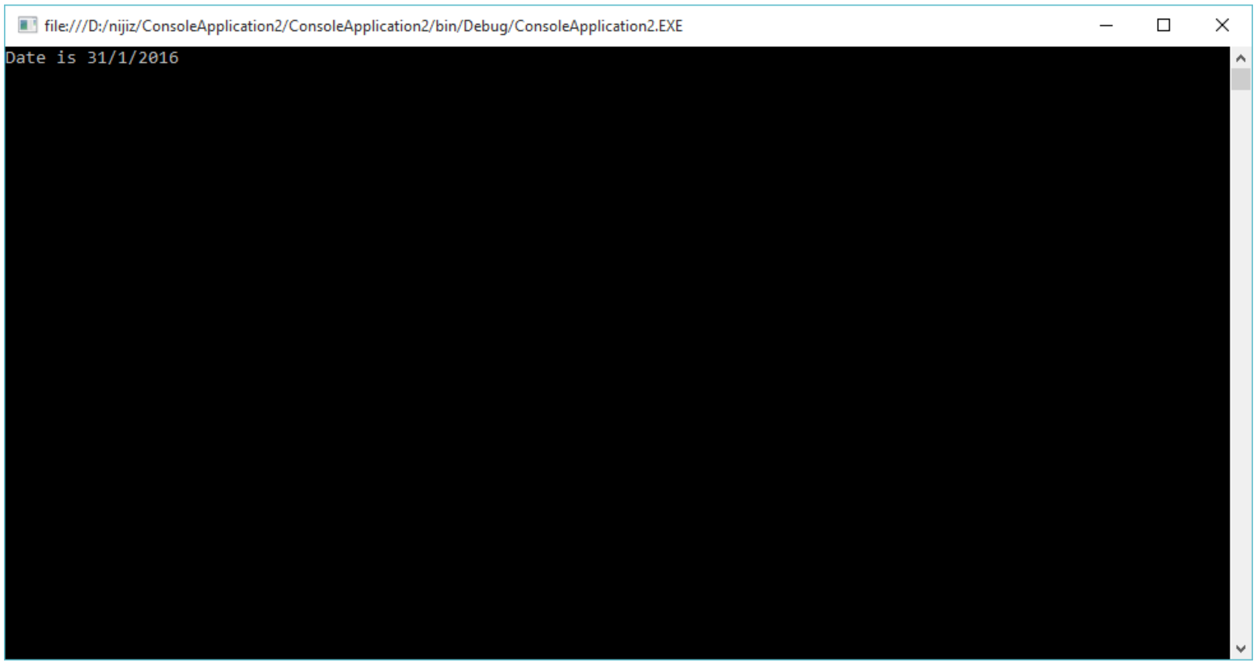
PROGRAM 3

TO IMPLEMENT INDEXERS

Declare a structure Date with Day, Month and Year. Declare an indexer for the structure Date so that DateObj[“Day”] returns the Day, DateObj[“Month”] returns the Month and DateObj[“Year”] returns the Year.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication2
{
    class MyDate
    {
        int Day;
        int Month;
        int Year;
          public int day { get { return Day; } }
          public int month { get { return Month; } }
          public int year { get { return Year; } }
          public MyDate(int a, int b, int c)
          {
              Day = a;
              Month = b;
              Year = c;
          }
          public void disp()
          {
            Console.WriteLine("Date :{0}/{1}/{2}", Day, Month, Year);
        }
    }
    class IndexerDemo
    {
        MyDate md;

        public int this[string s]
        {
            get
            {
                if (s == "Day") return md.day;
                else if (s == "Month") return md.month;
                else if (s == "Year") return md.year;
                else return 0;
            }
        }
        public IndexerDemo()
        {
            md = new MyDate(31, 1, 2016);
        }
    }
       class Program
       {
           public static void Main()
           {
               IndexerDemo ind = new IndexerDemo();
               Console.WriteLine("Date is {0}/{1}/{2}", ind["Day"], ind["Month"], ind["Year"]);
               Console.ReadKey();

           }
       }
   }
```

OUTPUT



file:///D:/nijiz/ConsoleApplication2/ConsoleApplication2/bin/Debug/ConsoleApplication2.EXE

```
Date is 31/1/2016
```

PROGRAM 4

TO IMPLEMENT THE CONCEPT OF INHERITANCE, ABSTRACT AND SEALED CLASSES

Declare a Person class with Name, DOB, and Address. Declare a derived class Student with
Person as the base class with course details as an abstract class. Declare a derived class
UGStudent with AdmnNo and Rank details as a sealed class. Declare appropriate functions to rad and display the details of 1 students.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication3b
    {
        public struct Date
        {
            int day;
            int month;
            int year;
            public Date(int d, int m, int y)
            {
                day = d; month = m; year = y;
            }
            public void read()
            {
                Console.Write("\n\nEnter Day:");
                day = Convert.ToInt32(Console.ReadLine());
                Console.Write("Enter Month:");
        month = Convert.ToInt32(Console.ReadLine());
         Console.Write("enter Year:");
         year = Convert.ToInt32(Console.ReadLine());
     }
     public void print()
     {
         Console.WriteLine("{0}/{1}/{2}", day, month, year);
     }
 }
 public abstract class Person
 {
     protected string firstname;
     protected string lastname;
     protected Date dob;
     protected string address;
     public Person()
     {

                         dob = new Date(0, 0, 0);
                         firstname = "N/A";
                         lastname = "N/A";
                         address = "N/A";
                     }
                     public Person(string fnm, string lnm, Date db, string addr)
                     {
                         dob = new Date();
                         firstname = fnm;
                         lastname = lnm;
                         dob = db;
                         address = addr;
                     }
                 }
                 public abstract class Student : Person
```

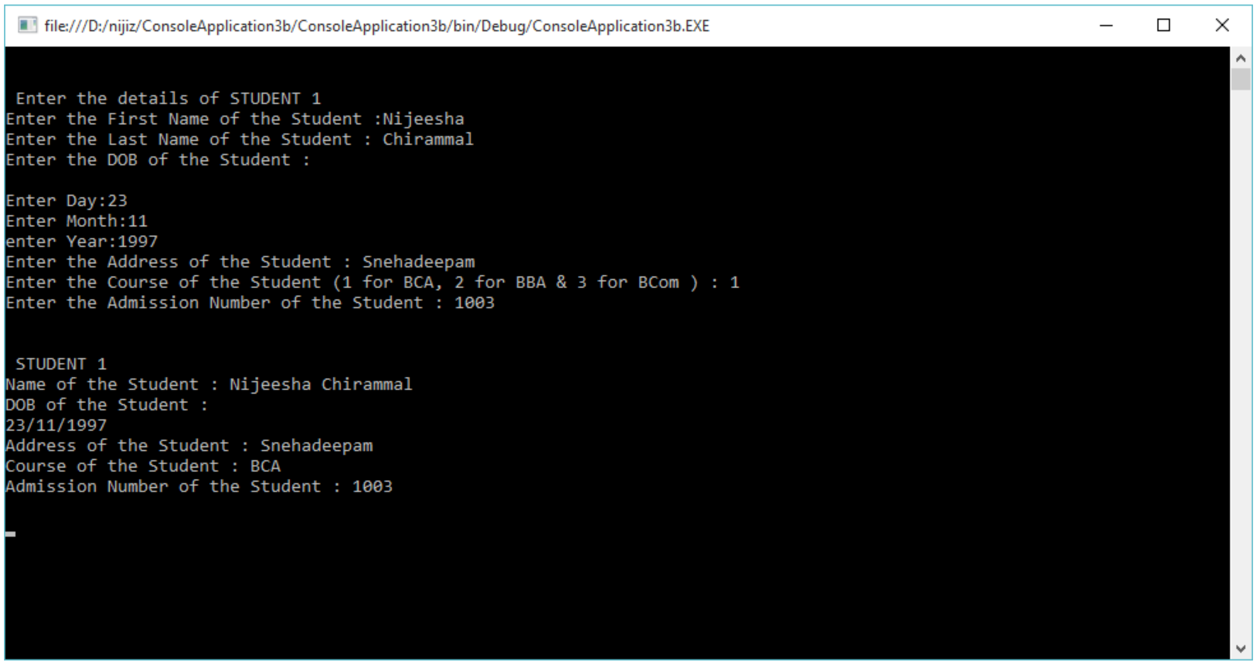
```
    {
  protected int admn_no;
  public Student()
  {
      dob = new Date();
      firstname = "N/A";
      lastname = "N/A";
      address = "N/A";
      admn_no = 0;
  }
  public Student(string fnm, string lnm, Date db, string addr, int adno)
      : base(fnm, lnm, db, addr)
  {
      admn_no = adno;
  }
}
public class UGStudent : Student
      {
          protected int course_code;
          public string course
          {
              set
              {
                  if (value == "BCA") course_code = 1;
                  else if (value == "BBA") course_code = 2;
                  else if (value == "Bcom") course_code = 3;
                  else course_code = 0;
              }
              get
              {
```

```
            if (course_code == 1) return "BCA";
            else if (course_code == 2) return "BBA";

        else if (course_code == 3) return "Bcom";
            else return "INVALID COURSE";
         }

    }
    public string name { get { return firstname + " " + lastname; } }

    public UGStudent()
    {
        dob = new Date(0, 0, 0);

        firstname = "N/A";

        lastname = "N/A";

        address = "N/A";
         course_code = 0;

         admn_no = 0;

     }

     public UGStudent(string fnm, string lnm, Date db, string addr, int adno, string

     cours)
         : base(fnm, lnm, db, addr, adno)
     {

         course = cours;
        }

        public void readdetails()
        {

            Console.Write("Enter the First Name of the Student :");

            firstname = Console.ReadLine();

            Console.Write("Enter the Last Name of the Student : ");

            lastname = Console.ReadLine();

            Console.Write("Enter the DOB of the Student : ");

            dob.read();

            Console.Write("Enter the Address of the Student : ");
              address = Console.ReadLine();

              Console.Write("Enter the Course of the Student (1 for BCA, 2 for BBA &3 for
    BCom ) : ");

              course_code = Convert.ToInt32(Console.ReadLine());

              Console.Write("Enter the Admission Number of the Student : ");

              admn_no = Convert.ToInt32(Console.ReadLine());

          }

          public void displaydetails()
          {
```

```
        Console.WriteLine("Name of the Student : {0}", name);

        Console.WriteLine("DOB of the Student :"); dob.print();

        Console.WriteLine("Address of the Student : {0}", address);

        Console.WriteLine("Course of the Student : {0}", course);

        Console.WriteLine("Admission Number of the Student : {0}", admn_no);

    }

}

public class InheritenceDemo
{
        public static void Main()
        {

            UGStudent[] studs = new UGStudent[10];

            for (int i = 0; i < 1; i++)
            {

                Console.WriteLine("\n\n Enter the details of STUDENT {0}", i + 1);

                studs[i] = new UGStudent();

                studs[i].readdetails();

            }
```

```
for (int i = 0; i < 1; i++)
            {
                Console.WriteLine("\n\n STUDENT {0}", i + 1);

                studs[i].displaydetails();

            }
            Console.WriteLine();
            Console.ReadKey();
        }

    }
}
```

OUTPUT



file:///D:/nijiz/ConsoleApplication3b/ConsoleApplication3b/bin/Debug/ConsoleApplication3b.EXE

```

 Enter the details of STUDENT 1
Enter the First Name of the Student :Nijeesha
Enter the Last Name of the Student : Chirammal
Enter the DOB of the Student :

Enter Day:23
Enter Month:11
enter Year:1997
Enter the Address of the Student : Snehadeepam
Enter the Course of the Student (1 for BCA, 2 for BBA & 3 for BCom ) : 1
Enter the Admission Number of the Student : 1003


 STUDENT 1
Name of the Student : Nijeesha Chirammal
DOB of the Student :
23/11/1997
Address of the Student : Snehadeepam
Course of the Student : BCA
Admission Number of the Student : 1003
```

PROGRAM 5

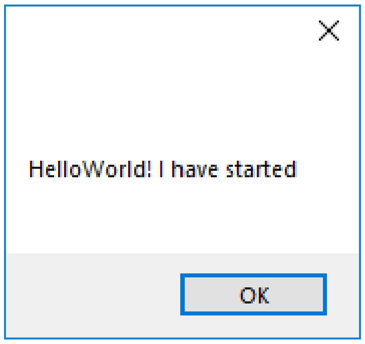
TO IMPLEMENT THE CONCEPT OF EVENTS AND DELEGATES

A simple program to display a message in the click event of a button

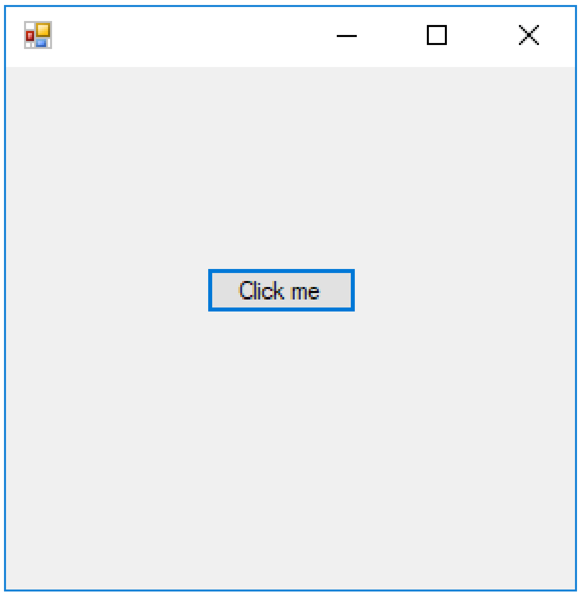
```
using System;
using System.Drawing;
using System.Windows.Forms;
//defining custom delegate
public delegate void EventDelegate();
public class HelloWorld : Form
{

    public event EventDelegate firstevent;
    static public void Main()
    {
       Application.Run(new HelloWorld());
    }
    public HelloWorld()
    {
       Button b = new Button();
       b.Text = "Click Me!";
       b.Location = new Point(100, 100);
       //using predefined EventHandler delegate
       b.Click += new EventHandler(Button_Click);
       Controls.Add(b);
       //using custm defined delegate
       firstevent = new EventDelegate(OnStartEvent);
       //trigger the event
       firstevent();
```

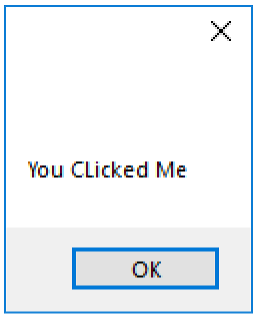
```
        }
        //Event handler method for cusom defined event
        public void OnStartEvent()
        {
            MessageBox.Show("\n\nHelloWorld ! I have Started !\n\n\n");
        }
        //Event handler method for predefined event (Click event)
        private void Button_Click(object sender, EventArgs e)
        {
            MessageBox.Show("\n\nYou Clicked Me!\n\n");
        }
    }
```

OUTPUT



```
HelloWorld! I have started
```





```
You CLicked Me
```

PROGRAM 6

TO IMPLEMENT THE CONCEPT OF DELEGATES

A program that declares a delegate called 'arithmetic_operation' that returns the result of the operation.

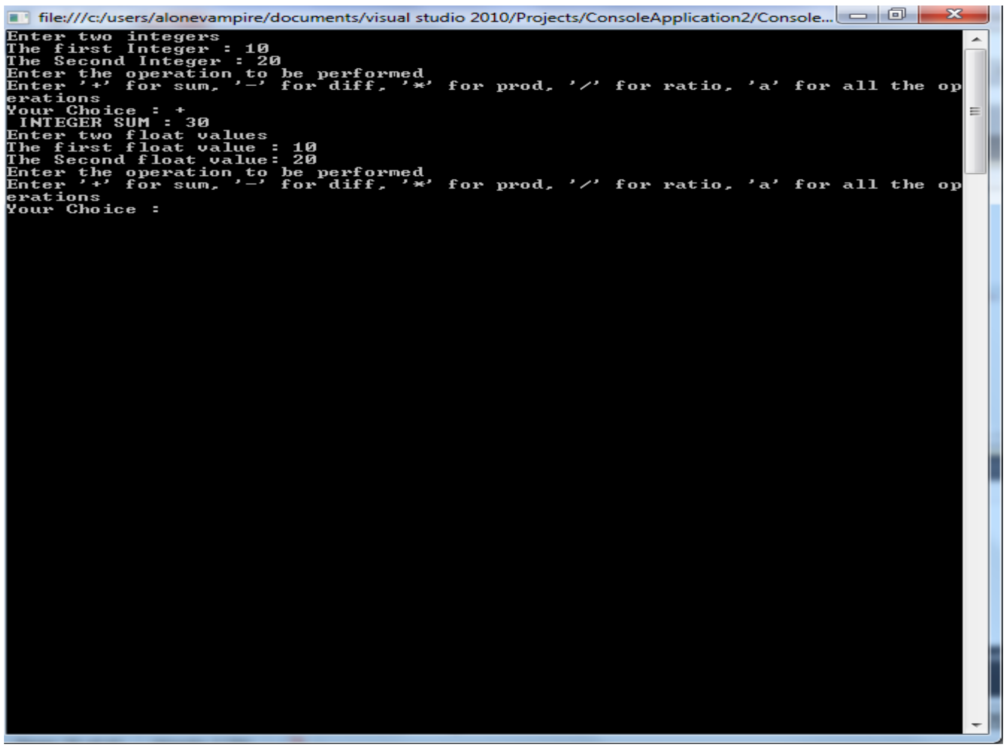
- **Four different operations add, sub, mul and div which can be invoked with delegate reference according to the user's choice of operation.**

```
using System;

public delegate void arithmetic_operation();

public class Integer

 {
 int v1,v2;
 public Integer(int x,int y){v1=x;v2=y;}
 public void Add(){ Console.WriteLine(" INTEGER SUM : {0}",v1+v2); } public void Sub()
 { Console.WriteLine(" INTEGER DIFF : {0}",v1-v2); } public void Mul()
 { Console.WriteLine(" INTEGER PROD : {0}",v1*v2); } public void Div()
 { Console.WriteLine(" INTEGER RATIO : {0}",v1/v2); }
 }
 public class Float
 {
 float f1,f2;
 public Float(float x, float y) { f1=x; f2=y;}
 public void Add(){ Console.WriteLine(" FLOAT SUM : {0}",f1+f2); }
 public void Sub(){ Console.WriteLine(" FLOAT DIFF : {0}",f1-f2); }
 public void Mul(){ Console.WriteLine(" FLOAT PROD : {0}",f1*f2); }
 public void Div(){ Console.WriteLine(" FLOAT RATIO : {0}",f1/f2); }
 }
 public class DelegateExample
 {

        public arithmetic_operation aop;
        public void operations(Integer ob, char opr)
        {
        if (opr== '+')
        aop = new arithmetic_operation(ob.Add);
        else if (opr =='-' )
        aop = new arithmetic_operation(ob.Sub);
        else if (opr== '*')

        aop = new arithmetic_operation(ob.Mul);
        else if (opr== '/')
        aop = new arithmetic_operation(ob.Div);

        else if (opr == 'a')

        aop = new arithmetic_operation(ob.Mul) + new arithmetic_operation(ob.Add)
```

```
+ new arithmetic_operation(ob.Sub) + new arithmetic_operation(ob.Div);
else aop=null;
aop();
}
public void operations(Float ob, char opr)
{
if (opr== '+')
aop = new arithmetic_operation(ob.Add);
else if (opr =='-' )
aop = new arithmetic_operation(ob.Sub);
else if (opr== '*')
aop = new arithmetic_operation(ob.Mul);
else if (opr== '/')
aop = new arithmetic_operation(ob.Div);
else if (opr == 'a')
aop = new arithmetic_operation(ob.Mul) + new arithmetic_operation(ob.Add)
+ new arithmetic_operation(ob.Sub) + new arithmetic_operation(ob.Div);
else aop=null;
aop();
}
}
public class ExampleProgram
{
public static void Main()
{
//create a delegate reference, 'de' an object with delegte reference 'aop' as its member
DelegateExample de = new DelegateExample();
//create an object of Integers class
Console.WriteLine("Enter two integers");
Console.Write("The first Integer : ");
int num1=Convert.ToInt32(Console.ReadLine());
Console.Write("The Second Integer : ");
int num2=Convert.ToInt32(Console.ReadLine());
```

```
Integer Intgr=new Integer(num1,num2);
Console.WriteLine("Enter the operation to be performed ");
Console.WriteLine("Enter '+' for sum, '-' for diff, '*' for prod, '/' for ratio, 'a' for all the
operations ");
Console.Write("Your Choice : ");
char c=Convert.ToChar(Console.ReadLine());
//accessing the methods with delegate reference for integer operations
//object de has the delegate reference 'aop' to invoke the methods
de.operations(Intgr,c);
//create an object of Float class
Console.WriteLine("Enter two float values");
Console.Write("The first float value : ");
float num3=Convert.ToSingle(Console.ReadLine());
Console.Write("The Second float value: ");
float num4=Convert.ToSingle(Console.ReadLine());
Float Flot=new Float(num3,num4);
Console.WriteLine("Enter the operation to be performed ");
Console.WriteLine("Enter '+' for sum, '-' for diff, '*' for prod, '/' for ratio, 'a' for all the
operations ");
Console.Write("Your Choice : ");
c=Convert.ToChar(Console.ReadLine());
//accessing the methods with delegate reference for float operations
//object de has the delegate reference 'aop' to invoke the methods
de.operations(Flot,c);
Console.ReadKey();
}
}
```

OUTPUT


file:///c:/users/alonevampire/documents/visual studio 2010/Projects/ConsoleApplication2/Console...

```
Enter two integers
The first Integer : 10
The Second Integer : 20
Enter the operation to be performed
Enter '+' for sum, '-' for diff, '*' for prod, '/' for ratio, 'a' for all the op
erations
Your Choice : +
 INTEGER SUM : 30
Enter two float values
The first float value : 10
The Second float value: 20
Enter the operation to be performed
Enter '+' for sum, '-' for diff, '*' for prod, '/' for ratio, 'a' for all the op
erations
Your Choice :
```

PROGRAM 7

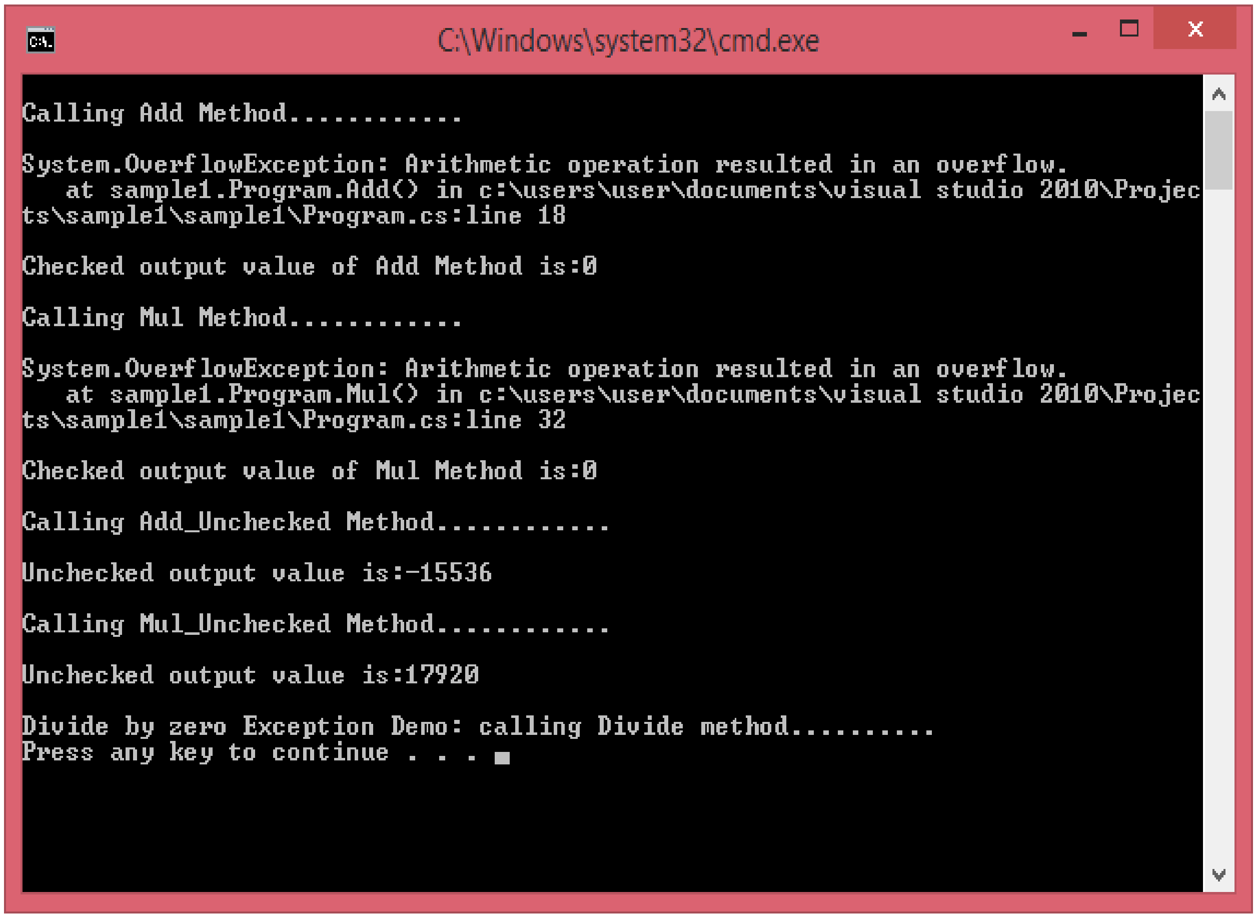
TO IMPLEMENT THE CONCEPT OF EXCEPTION HANDLING

Write a program that handles a divide by zero error and also arithmetic overflow error
With 'checked' and 'unchecked' statements

```
using System;

class ExceptionProgram
{

    public short a = 30000; public short b = 20000; public short c;

    // Add and Mul methods use checked operator

    public int Add()
    {

        try
        {

            c = checked((short)(a + b));

        }

        catch (System.OverflowException e)
        {
           System.Console.WriteLine("\n" + e.ToString());
        }
         return c;
```

```
}

public int Mul()
{
   try
       {
           checked
           {
               c = (short)(a * b);
           }
       }

       catch (System.OverflowException e)
       {
          System.Console.WriteLine("\n" + e.ToString());
        }
        return c;

    }

    // Add_Unchecked and Mul_Unchecked methods use unchecked operator

    public int Add_Unchecked()
    {

        try
        {

            c = unchecked((short)(a + b));
        }
        catch (System.OverflowException e)
        {

            System.Console.WriteLine("\n" + e.ToString());

        }

        return c;

    }

    public int Mul_Unchecked()
    {

        try
        {

            unchecked
            {

                c = (short)(a * b);

            }

        }

        catch (System.OverflowException e)
        {

            System.Console.WriteLine("\n" + e.ToString());

        }
        return c;

    }

    public void Divide()
    {

        string msg = null;

        int c1 = 999999, c2 = 999999;

        try
        {

            c1 = a / b;

            b = 0;

            c2 = a / b;

        }

        catch (DivideByZeroException e)
        {

            msg = "\n Divide By Zero Eeception Occurred ! \n " + e.ToString();

        }

        finally
        {
            if (msg == null)

                Console.WriteLine("\n Excption has not occurred msg ==> {0}", msg);

            else

                Console.WriteLine("\n Excption ==> {0}", msg);

            Console.WriteLine("\nOutput of Division operation : c1 = {0}, c2= {1} ", c1, c2);

        }

    }

    public static void Main(string[] args)
    {

        ExceptionProgram p = new ExceptionProgram();

        Console.WriteLine("\nCalling Add method ....................");

        Console.WriteLine("\n Checked output value of Add method is: {0}", p.Add());

        Console.WriteLine("\nCalling Mul method ....................");

        Console.WriteLine("\n Checked output value of Mul method is: {0}", p.Mul());

        Console.WriteLine("\nCalling Add_Unchecked method ....................");

        Console.WriteLine("\n UnChecked output value is: {0}", p.Add_Unchecked());

        Console.WriteLine("\nCalling Mul_Unchecked method ....................");
        Console.WriteLine("\n UnChecked output value is: {0}", p.Mul_Unchecked());

        //For Divide Overflow error

        Console.WriteLine("\nDivide by Zero Exception Demo : Calling Divide method
..........");

        p.Divide();

    }

}
```

OUTPUT



```
Calling Add Method............

System.OverflowException: Arithmetic operation resulted in an overflow.
   at sample1.Program.Add() in c:\users\user\documents\visual studio 2010\Projec
ts\sample1\sample1\Program.cs:line 18

Checked output value of Add Method is:0

Calling Mul Method............

System.OverflowException: Arithmetic operation resulted in an overflow.
   at sample1.Program.Mul() in c:\users\user\documents\visual studio 2010\Projec
ts\sample1\sample1\Program.cs:line 32

Checked output value of Mul Method is:0

Calling Add_Unchecked Method............

Unchecked output value is:-15536

Calling Mul_Unchecked Method............

Unchecked output value is:17920

Divide by zero Exception Demo: calling Divide method..........
Press any key to continue . . .
```

PROGRAM:8

TO IMPLEMENT WINDOWS FORM APPLICATION

To design a calculator in windows form

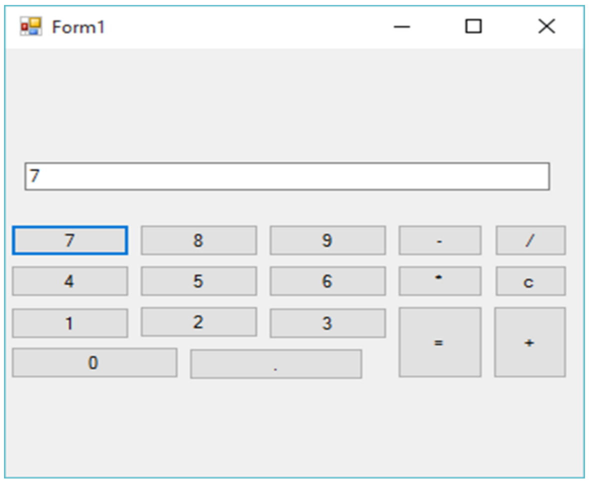
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace c
{
    public partial class Form1 : Form
    {
        String s=null;
        double var1,var2;
        String sign;
        double res=0;


        public Form1()
        {
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {

            textBox1.Text = s + "2";
    s = s + "2";
}

private void button6_Click(object sender, EventArgs e)
{

    textBox1.Text = s + "6";
    s = s + "6";
}

private void Form1_Load(object sender, EventArgs e)
{

}

private void button1_Click(object sender, EventArgs e)
{
   textBox1.Text=s+"1";
   s = s + "1";
}

private void textBox1_TextChanged(object sender, EventArgs e)
{

}

private void button3_Click(object sender, EventArgs e)
{

    textBox1.Text = s + "3";
    s = s + "3";
}
```

```
private void button4_Click(object sender, EventArgs e)
{
    textBox1.Text = s + "4";
    s = s + "4";
}

private void button5_Click(object sender, EventArgs e)
{

    textBox1.Text = s + "5";
    s = s + "5";
}

private void button7_Click(object sender, EventArgs e)
{

    textBox1.Text = s + "7";
    s = s + "7";
}

private void button8_Click(object sender, EventArgs e)
{

    textBox1.Text = s + "8";
    s = s + "8";
}

private void button9_Click(object sender, EventArgs e)
{

    textBox1.Text = s + "9";
    s = s + "9";
}

private void button10_Click(object sender, EventArgs e)
{

    textBox1.Text = s + "0";
    s = s + "0";
}

private void button11_Click(object sender, EventArgs e)
{

    sign = "+";
    var1 = Convert.ToDouble(s);
    s = "";
    textBox1.Text = "";
}

private void button12_Click(object sender, EventArgs e)
{
    sign = "-";
    var1 = Convert.ToDouble(s);
    s = "";
    textBox1.Text = "";
}

private void button13_Click(object sender, EventArgs e)
{
    sign = "*";
    var1 = Convert.ToDouble(s);
    s = "";
    textBox1.Text = "";
}
```

```
private void button14_Click(object sender, EventArgs e)
{
    sign = "/";
    var1 = Convert.ToDouble(s);
    s = "";
    textBox1.Text = "";
}

private void button15_Click(object sender, EventArgs e)
{
    textBox1.Text = s + ".";
    s = s + ".";
}

private void button16_Click(object sender, EventArgs e)
{
    var2 = Convert.ToDouble(s);
    s = "";
    if (sign == "+")
    {
        res = var1 + var2;
        textBox1.Text = res.ToString();
    }
    else if (sign == "-")
    {
        res = var1 - var2;
        textBox1.Text = res.ToString();
    }
     else if(sign == "*")
    {
        res = var1 * var2;
        textBox1.Text = res.ToString();
```

```
            }
            else if (sign == "/")
            {
                res = var1 / var2;
                textBox1.Text = res.ToString();
            }
            s = res.ToString();
        }

        private void button17_Click(object sender, EventArgs e)
        {
            s = "";
            res = 0;
            textBox1.Text = "";
            var1 = 0;
            var2 = 0;
        }
    }
}
```

OUTPUT



PROGRAM NO:9

To implement validation control in web form