

**C# & .NET FRAMEWORK LAB MANUAL**

**Department of Computer Science and  
Engineering**

**C# & .NET FRAMEWORK LAB MANUAL**

**Lab Manual for the Academic Year 2020-2021  
III B.Tech IIsem**



**Rajeev Gandhi Memorial College Of  
Engineering and  
Technology(Autonomous)  
Nandyal, Kurnool District – 518 501 (A. P.)**

**In-charge**

**G. RAJA SEKHAR REDDY**



**VISION OF THE DEPARTMENT**

- To empower students with cutting edge technologies in computer science and engineering
- To train the students as entrepreneurs in computer science and engineering to address the needs of the society
- To develop smart applications to disseminate information to rural people

**MISSION OF THE DEPARTMENT**

- To become the best computer science and engineering department in the region offering undergraduate, post graduate and research programs in collaboration with industry
- To incubate, apply and spread innovative ideas by collaborating with relevant industries and R & D labs through focused research groups.
- To provide exposure to the students in the latest tools and technologies to develop smart applications for the society

# C# & .NET FRAMEWORK LAB MANUAL

III B. Tech. II- Sem (CSE)

Practical Credits

3

2

(A0594155) C# & .NET FRAMEWORK LAB

(COMMON TO CSE & IT)

## OBJECTIVE:

- ❖ The student will gain knowledge in the concepts of the .NET framework as a whole and the technologies that constitute the Framework.
- ❖ The student will gain programming skills in C# both in basic and advanced levels.
- ❖ By building sample applications, the student will get experience and be ready for large-scale projects.

## OUTCOMES:

- ❖ Write, compile and debug programs in C# language.
- ❖ To learn the basics of object oriented programming.
- ❖ To get knowledge of windows programming.
- ❖ To get knowledge on server side programming.
- ❖ To gain knowledge on web services and dynamic link libraries.

## CO-PO MAPPING:

CO /PO	P O 1	P O 2	P O 3	P O 4	P O 5	P O 6	P O 7	P O 8	P O 9	P O 10	P O 11	P O 12	PS O 1	PS O 2	PS O 3
CO1	1	1	1	1	2	1			1	1		2			
CO2	2	2	1	1	2				1		1	2			
CO3	2	2	2	2	2	1			2			2			
CO4	2	2	1	1	3				1	1		2			
CO5	2	2	2	2	2	1			2			1			

# **C# & .NET FRAMEWORK LAB MANUAL**

## **LAB-1**

1. Write a program to C# to find the smallest single digit factor for a given value.
2. Write a program in C# to print a number if it is prime; otherwise display the largest factor of that number.
3. Write a program in C# to find the magnitude of a number.

## **LAB-2**

1. Write a C# program for addition and multiplication of two matrices.
2. Write a C# program to display the digits of an integer in words.
3. Write a C# program to which reads a set of strings into the rows a two dimensional array and then prints the string having more number of vowels.

## **LAB-3**

1. Write a C# programs to demonstrate the concepts of Structures and Enumerations.
2. Write a C# programs to demonstrate the concepts of Constructors and Inheritance.
3. Write a C# programs to demonstrate the concepts of Polymorphism.

## **LAB-4**

1. Write a C# programs to demonstrate the concepts of Partial classes and Extension methods.
2. Write a C# programs to demonstrate the concepts of Delegates.

## **LAB-5**

1. Write a C# programs to demonstrate the concepts of Label, Text Box and Button controls.
2. Write a C# programs to demonstrate the concepts of Combo Box and List Box controls.

# C# & .NET FRAMEWORK LAB MANUAL

## **LAB-6**

1. Create a Windows application in C# for registration form and fill the details and when you click the submit button it display the details in the message box.
2. Create a Windows application in C# having two text boxes and three buttons named as factorial, prime, factorial series. When you click any button the resultant value will be displayed on the second textbox.

## **LAB-7**

1. Create a ADO.NET application in C# to verify if the connection is established with OLEDB and MS-ACCESS.
2. Create a ADO.NET applications in C# to demonstrate the Data Reader, Data Set, Data Adapter and Data View Objects.

## **LAB-8**

1. Develop the Static Web pages using HTML and some validations along with Java Script.
2. Design an ASP.NET Webpage to demonstrate the Label, Button and Textbox controls.

## **LAB-9**

1. Design an ASP.NET Webpage to work with Cross page and Post back Submissions.
2. Design an ASP.NET Webpage to work with Dropdown list and ListBox controls.

## **LAB-10**

1. Develop a Registration Form with all Validation Controls.
2. Create a Web Service for all Arithmetic operations.

# C# & .NET FRAMEWORK LAB MANUAL

## **LAB-11**

1. Write a C# program to implement Assemblies.
2. Write a C# program to implement Multithreading and Thread Synchronization.

## **REFERENCES:**

1. Programming in C#, E. Balagurusamy, Tata McGraw-Hill, 2004.
2. Programming C#, J. Liberty, 2<sup>nd</sup> Edition., O'Reilly, 2002.
3. C# and the .NET Platform, Andrew Trolesen, 2<sup>nd</sup> Edition, Dreamtech Press
4. Sams Teach Yourself the C# Language in 21 Days', Bradley L Jones, 1<sup>st</sup> edition, 2001.

# C# & .NET FRAMEWORK LAB MANUAL

## LAB1:

1. Write a program to C# to find the smallest single digit factor for a given value.

```
using System;
namespace Factor
{
    class Program
    {
        public static void main( )
        {
            int i=2,n;
            Console.WriteLine("enter n value");
            n=int.Parse(Console.ReadLine);
            while(i<=n)
            {
                if(n<10&&i<=n)
                {
                    Console.WriteLine(n+"doesn't have single
                        digit factor other than one");
                    break;
                }
                if(n%i==0)
                {
                    Console.WriteLine(i+"is smallest factor");
                    break;
                }
                i++;
            }
            Console.ReadKey();
        }
    }
}
```

## C# & .NET FRAMEWORK LAB MANUAL

**2. Write a program in C# to print a number if it is prime; otherwise display the largest factor of that number.**

```
using System;
namespace prime
{
    class Pro
    {
        public static void main()
        {
            int i,c,m=1;
            Console.WriteLine("Enter a value");
            int n=int.Parse(Console.ReadLine());
            c=0;i=1;
            while(i<=n)
            {
                if(n%i==0)
                {
                    if(i!=n)
                    m=i;
                    c++;
                }
                i++;
            }
            if(c==2)
                Console.WriteLine("num is prime");
            else
                Console.WriteLine(m+ "is largest factor for "+n);
            Console.ReadKey();
        }
    }
}
```



# C# & .NET FRAMEWORK LAB MANUAL

## 3. Write a program in C# to find the magnitude of a number.

using System;

namespace c

{

    class pro

    {

        public static void main(String[] args)

        {

            int n,i=0;

            Console.WriteLine("enter n value");

            int n=int.Parse(Console.ReadLine());

            while(n>0)

            {

                i++;

                n=n/10;

            }

            Console.WriteLine("Magnitude of given num is:"+i);

        }

    }

}

# C# & .NET FRAMEWORK LAB MANUAL

## Lab-2:

### 1. Write a C# program for addition and multiplication of two matrices.

using System;

namespace matrix

```
{
    class Pro1
    {
        public static void main(String[] args)
        {
            int i,j,m,n,p,q;
            int [,]a=new int[30,30];
            int [,]b=new int[30,30];
            int[,]c=new int[30,30];
            Console.WriteLine("enter rows and columns of first
                               matrix");
            m=Convert.ToInt32(Console.ReadLine());
            n= Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("enter rows and columns of second
                               matrix");
            p=Convert.ToInt32(Console.ReadLine());
            q= Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Elemnts of first matrix");
            for(i=0;i<m;i++)
            {
                for(j=0;j<n;j++)
                    a[i,j]=int.Parse(Console.ReadLine());
            }
            for(i=0;i<p;i++)
            {
                for(j=0;j<q;j++)
                    b[i,j]=int.Parse(Console.ReadLine());
            }
            if((m==p) && (n==q))
```

## C# & .NET FRAMEWORK LAB MANUAL

```
{
    Console.WriteLine("addition is possible");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
            c[I,j]=a[I,j]+b[I,j];
    }
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
            Console.WriteLine(c[I,j]+"/t");
        Console.WriteLine("/n");
    }
}
else
    Console.WriteLine("addition is not possible");
if(n!=p)
    Console.WriteLine("Mul is not possible");
else
{
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            c[I,j]=0;
            for(k=0;k<m&& k<q;k++)
                c[I,j]=c[I,j]+a[I,k]*b[k,j];
        }
    }
    Console.WriteLine("The matrix is");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
```

# C# & .NET FRAMEWORK LAB MANUAL

```
        Console.WriteLine("\t"+c[I,j]);
    Console.WriteLine("\n");
    }
}
}
}
```

# C# & .NET FRAMEWORK LAB MANUAL

## 2. Write a C# program to display the digits of an integer in words.

using System;

namespace Application

{

class Program

{

public static void main(String[] args)

{

int num,nextdigit,numdigits;

int[] n=new int[20];

string[] digits={"zero","one","two","three","four","five","six",  
"seven","eight","nine"};

Console.WriteLine("enter the number");

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

Console.WriteLine("number"+num);

Console.WriteLine("number in words");

nextdigit=0;

numdigits=0;

do

{

nextdigits=num%10;

n[numdigits]=nextdigit;

numdigits++;

num=num/10;

}while(num>0);

numdigits--;

for( ;numdigits>=0;numdigits--)

Console.WriteLine(digits[n[numdigits]]+" ");

Console.ReadLine();

}

}

}

## C# & .NET FRAMEWORK LAB MANUAL

**3. Write a C# program to which reads a set of strings into the rows a two dimensional array and then prints the string having more number of vowels.**

```
using System;
using System.Collection.Generic;
using System.Linq;
using System.Text;
namespace Program
{
    class B
    {
        public static void main(String[] args)
        {
            string[,]a =new String[10,10];
            char[] x=new Char[10];
            int[] t=new int[10];
            string[] g=new String[10];
            string b;
            Console.WriteLine("enter rows");
            int m=int .Parse(Console.ReadLine());
            Consoel.WriteLine("enter coloumns");
            int n=int.Parse("Console.WriteLine());
            for(int i=0;i<m;i++)
            {
                for(int j=0;j<n;j++)
                    a[I,j]=Console.ReadLine();
            }
            int k=0,c=0,s=0;
            for(int i=0;i<m;i++)
            {
                for(int j=0;j<n;j++)
                {
                    b=a[I,j];
```

## C# & .NET FRAMEWORK LAB MANUAL

```
        c=0;
        for(k=0;k<b.Length;k++)
        {
            if(b[k]=='a')
                c++;
            else if(b[k]=='e')
                c++;
            else if(b[k]=='I')
                c++;
            else if(b[k]=='o')
                c++;
            else if(b[k]=='u')
                c++;
        }
        g[s]=b;
        t[s]=c;
        s++;
    }
}
int temp;
string tem;
for(int j=0;j<s;j++)
{
    for(int i=0;i<s;i++)
    {
        if(t[i]>=t[j])
        {
            temp=t[i];
            t[i]=t[j];
            t[j]=temp;
            temp=g[i];
            g[i]=g[j];
            g[j]=temp;
        }
    }
}
```

## C# & .NET FRAMEWORK LAB MANUAL

```
        }
    }
}
Console.WriteLine("strings having more numbers of
                    vowels are");
for(int i=0;i<s;i++)
{
    if(t[i]<=t[s-1])
        Console.WriteLine(g[i]);
}
Console.WriteLine("word has "+t[s-1]+"vowels);
Console.ReadKey();
}
}
}
```



# C# & .NET FRAMEWORK LAB MANUAL

## Lab-3

### 1. Write a C# programs to demonstrate the concepts of Structures.

using System;

```
struct Books
```

```
{
```

```
    public string title, author, subject;
```

```
    public int book_id;
```

```
};
```

```
public class testStructure
```

```
{
```

```
    public static void Main(string[] args)
```

```
    {
```

```
        Books Book1;
```

```
        Books Book2;
```

```
        Book1.title = "C Programming";
```

```
        Book1.author = "Nuha Ali";
```

```
        Book1.subject = "C Programming Tutorial";
```

```
        Book1.book_id = 6495407;
```

```
        Book2.title = "Telecom Billing";
```

```
        Book2.author = "Zara Ali";
```

```
        Book2.subject = "Telecom Billing Tutorial";
```

```
        Book2.book_id = 6495700;
```

```
        Console.WriteLine("Book 1 title : {0}", Book1.title);
```

```
        Console.WriteLine("Book 1 author : {0}", Book1.author);
```

```
        Console.WriteLine("Book 1 subject : {0}", Book1.subject);
```

```
        Console.WriteLine("Book 1 book_id :{0}", Book1.book_id);
```

```
        Console.WriteLine("Book 2 title : {0}", Book2.title);
```

```
        Console.WriteLine("Book 2 author : {0}", Book2.author);
```

```
        Console.WriteLine("Book 2 subject : {0}", Book2.subject);
```

```
        Console.WriteLine("Book 2 book_id : {0}", Book2.book_id);
```

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

```
    }
```

```
}
```

# C# & .NET FRAMEWORK LAB MANUAL

**Write a C# programs to demonstrate the concepts of Enumerations.**

using system

class Enumtype

```
{  
    enum Direction  
    {  
        North,  
        East==10,  
        West,  
        South  
    }  
    public static void Main()  
    {  
        Direction d1=0,  
        Direction d2= Direction.East;  
        Direction d3= Direction.West;  
        Direction d4= (Direction)12;  
        Console.WriteLine("d1="+d1);  
        Console.WriteLine("d2="+(int)d2);  
        Console.WriteLine("d3="+d3);  
        Console.WriteLine("d4="+d4);  
    }  
}
```

# C# & .NET FRAMEWORK LAB MANUAL

## 2. Write a C# programs to demonstrate the concepts of Default Constructors

```
using System;
namespace ConsoleApplication3
{
    class Sample
    {
        public string param1, param2;
        public Sample()    // Default Constructor
        {
            param1 = "Welcome";
            param2 = "Aspdotnet-Suresh";
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Sample obj=new Sample();
            Console.WriteLine(obj.param1);
            Console.WriteLine(obj.param2);
            Console.ReadLine();
        }
    }
}
```

# C# & .NET FRAMEWORK LAB MANUAL

**Write a C# programs to demonstrate the concepts of Parameterized Constructors**

```
using System;
namespace ConsoleApplication3
{
    class Sample
    {
        public string param1, param2;
        public Sample(string x, string y
        {
            param1 = x;
            param2 = y;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Sample obj=new Sample("Welcome","Asp.Net");
            Sample obj1=new Sample("Welcome1","Asp.Net");
            Console.WriteLine(obj.param1 +" to "+ obj.param2);
            Console.WriteLine(obj1.param1 +" to "+ obj1.param2);
            Console.ReadLine();
        }
    }
}
```

# C# & .NET FRAMEWORK LAB MANUAL

**Write a C# programs to demonstrate the concepts of copy Constructors**

using System;

namespace ConsoleApplication3

```
{  
    class Sample  
    {  
        public string param1, param2;  
        public Sample(string x, string y)  
        {  
            param1 = x;  
            param2 = y;  
        }  
        public Sample(Sample obj)    // Copy Constructor  
        {  
            param1 = obj.param1;  
            param2 = obj.param2;  
        }  
    }  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Sample obj = new Sample("Welcome", "Asp.Net");  
            Sample obj1=new Sample(obj);  
            Console.WriteLine(obj1.param1 +" to " + obj1.param2);  
            Console.ReadLine();  
        }  
    }  
}
```

# C# & .NET FRAMEWORK LAB MANUAL

**Write a C# programs to demonstrate the concepts of static Constructors**

using System;

namespace ConsoleApplication3

```
{
    class Sample
    {
        public string param1, param2;
        static Sample()
        {
            Console.WriteLine("Static Constructor");
        }
        public Sample()
        {
            param1 = "Sample";
            param2 = "Instance Constructor";
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Sample obj=new Sample();
            Console.WriteLine(obj.param1 + " " + obj.param2);
            Sample obj1 = new Sample();
            Console.WriteLine(obj1.param1 + " " + obj1.param2);
            Console.ReadLine();
        }
    }
}
```

# C# & .NET FRAMEWORK LAB MANUAL

**Write a C# programs to demonstrate the concepts of private Constructors**

using System;

namespace ConsoleApplication3

```
{
    public class Sample
    {
        public string param1, param2;
        public Sample(string a,string b)
        {
            param1 = a;
            param2 = b;
        }
        private Sample() // Private Constructor Declaration
        {
            Console.WriteLine("Private Constructor with no pramete");
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Sample obj = new Sample("Welcome","to ASP.NET");
            Console.WriteLine(obj.param1 + " " + obj.param2);
            Console.ReadLine();
        }
    }
}
```

# C# & .NET FRAMEWORK LAB MANUAL

**Write a C# programs to demonstrate the concepts of single Inheritance**

using System;

namespace InheritanceApplication

```
{
    class a
    {
        public void display()
        {
            System.Console.WriteLine("hahahaha");
        }
    }
    class b : a //b is child of a
    {
        public void display1()
        {
            System.Console.WriteLine("hihihih");
        }
    }
    class c
    {
        public static void Main()
        {
            b x=new b();//Normally object of child
            x.display();
            x.display1();
        }
    }
}
```



# C# & .NET FRAMEWORK LAB MANUAL

**Write a C# programs to demonstrate the concepts of Hierarchical Inheritance**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Inheritance
{
    class Program
    {
        static void Main(string[] args)
        {
            Principal g = new Principal();
            g.Monitor();
            Teacher d = new Teacher();
            d.Monitor();
            d.Teach();
            Student s = new Student();
            s.Monitor();
            s.Learn();
            Console.ReadKey();
        }
    }
    class Principal
    {
        public void Monitor()
        {
            Console.WriteLine("Monitor");
        }
    }
    class Teacher : Principal
    {
        public void Teach()
        {
```

## C# & .NET FRAMEWORK LAB MANUAL

```
        Console.WriteLine("Teach");
    }
}
class Student : Principal
{
    public void Learn()
    {
        Console.WriteLine("Learn");
    }
}
}
```

# C# & .NET FRAMEWORK LAB MANUAL

**Write a C# programs to demonstrate the concepts of multilevel Inheritance**

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
namespace ConsoleApplication8
```

```
{
```

```
    class a
```

```
    {
```

```
        public void display()
```

```
        {
```

```
            Console.WriteLine("hahahaha");
```

```
        }
```

```
    }
```

```
    class b : a //b is child of a
```

```
    {
```

```
        public void display1()
```

```
        {
```

```
            Console.WriteLine("hihihih");
```

```
        }
```

```
    }
```

```
    class d : b //d is child of b
```

```
    {
```

```
        public void display2()
```

```
        {
```

```
            Console.WriteLine("hohohohoh");
```

```
        }
```

```
    }
```

```
    class c
```

```
    {
```

```
        public static void Main()
```

```
        {
```

## C# & .NET FRAMEWORK LAB MANUAL

```
        d x = new d();//Normally object of child
        x.display();
        x.display1();
        x.display2();
    }
}
```

## C# & .NET FRAMEWORK LAB MANUAL

**3. Write a C# programs to demonstrate the concepts of Polymorphism.  
Write a C# programs to demonstrate the concepts of method overloading.**

```
using System;
```

```
namespace PolymorphismApplication
```

```
{
```

```
    class Printdata
```

```
    {
```

```
        void print(int i)
```

```
        {
```

```
            Console.WriteLine("Printing int: {0}", i );
```

```
        }
```

```
        void print(double f)
```

```
        {
```

```
            Console.WriteLine("Printing float: {0}" , f);
```

```
        }
```

```
        void print(string s)
```

```
        {
```

```
            Console.WriteLine("Printing string: {0}", s);
```

```
        }
```

```
        static void Main(string[] args)
```

```
        {
```

```
            Printdata p = new Printdata();
```

```
            p.print(5);
```

```
            p.print(500.263);
```

```
            p.print("Hello C++");
```

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

```
        }
```

```
    }
```

```
}
```

# C# & .NET FRAMEWORK LAB MANUAL

**Write a C# programs to demonstrate the concepts of constructor overloading.**

using System;

namespace ConsoleApplication3

{

    class Sample

    {

        public string param1, param2;

        public Sample() // Default Constructor

        {

            param1 = "Hi";

            param2 = "I am Default Constructor";

        }

        public Sample(string x, string y)

        {

            param1 = x;

            param2 = y;

        }

    }

    class Program

    {

        static void Main(string[] args)

        {

            Sample obj = new Sample();

            Sample obj1=new Sample("Welcome","ASP.NET");

            Console.WriteLine(obj.param1 + ", "+obj.param2);

            Console.WriteLine(obj1.param1 +" to " + obj1.param2);

            Console.ReadLine();

        }

    }

## C# & .NET FRAMEWORK LAB MANUAL

**Write a C# programs to demonstrate the concepts of unary operator overloading.**

```
using System;
class bank
{
    int x;
    int y;
    public bank(int a, int b)
    {
        x = a;
        y = b;
    }
    public bank()
    {
    }
    public void display()
    {
        Console.Write(" " + x);
        Console.Write(" " + y);
        Console.WriteLine();
    }
    public static bank operator -(bank b)
    {
        b.x = -b.x;
        b.y = -b.y;
        return b;
    }
}
class program
{
    public static void Main()
    {
        bank ba1 = new bank(10,-20);
```

## C# & .NET FRAMEWORK LAB MANUAL

```
        ba1.display();
        bank ba2 = new bank();
        ba2.display();
        ba2 = -ba1;
        ba2.display();
        Console.ReadLine();
    }
}
```



# C# & .NET FRAMEWORK LAB MANUAL

**Write a C# programs to demonstrate the concepts of binary operator overloading.**

```
using System;
namespace binary_overload
{
    class complexNumber
    {
        int x;
        double y;
        public complexNumber(int real, double imaginary)
        {
            x = real;
            y = imaginary;
        }
        public complexNumber()
        {
        }
        public static complexNumber operator +(complexNumber c1,
            complexNumber c2)
        {
            complexNumber c = new complexNumber();
            c.x=c1.x+c2.x;
            c.y=c1.x-c2.y;
            return c;
        }
        public void show()
        {
            Console.Write(x);
            Console.Write("+j"+y);
            Console.WriteLine();
        }
    }
    class Program
```

## C# & .NET FRAMEWORK LAB MANUAL

```
{
static void Main(string[] args)
{
    complexNumber p, q, r;
    p = new complexNumber(10, 2.0);
    q = new complexNumber(20, 15.5);
    r = p + q;
    Console.Write("p=");
    p.show();
    Console.Write("q=");
    q.show();
    Console.Write("r=");
    r.show();
    Console.ReadLine();
}
}
```

# C# & .NET FRAMEWORK LAB MANUAL

**Write a C# programs to demonstrate the concepts of comparison operator overloading.**

using System;

namespace comparison

```
{
    class Vector
    {
        int x, y, z;
        public Vector(int p, int q, int r)
        {
            x = p;
            y = q;
            z = r;
        }
        public static bool operator ==(Vector v1, Vector v2)
        {
            if (v1.x == v2.x && v1.y == v2.y && v1.z == v2.z)
                return (true);
            else
                return (false);
        }
        public static bool operator !=(Vector v1, Vector v2)
        {
            return (!(v1 == v2));
        }
    }
    class comparison
    {
        static void Main()
        {
            Vector v1 = new Vector(10, 20, 30);
            Vector v2 = new Vector(40, 50, 60);
            if (v1 == v2)
```

## C# & .NET FRAMEWORK LAB MANUAL

```
        Console.WriteLine("v1 and v2 both are Equal");
    else
        Console.WriteLine("v1 and v2 are not equal");
    if (!(v1 == v2))
        Console.WriteLine("true");
    else
        Console.WriteLine("false");
    Console.ReadLine();
}
}
}
```

# C# & .NET FRAMEWORK LAB MANUAL

## LAB-4:

### 1. Write a C# programs to demonstrate the concepts of Partial classes

using System;

namespace App

```
{
    public partial class partialclass
    {
        private int x;
        private int y;
        public partialclass(int x,int y)
        {
            this.x=x;
            this.y=y;
        }
    }
    public partial class partialclass
    {
        public void print()
        {
            Console.WriteLine("output values{0},{1}", x, y);
        }
    }
    class test
    {
        public static void Main()
        {
            partialclass p=new partialclass(10,15);
            p.print();
            Console.ReadLine();
        }
    }
}
```

# C# & .NET FRAMEWORK LAB MANUAL

## 2. Write a C# programs to demonstrate the concepts of single cast Delegates.

```
using System;
delegate int NumberChanger(int n);
namespace DelegateAppl
{
    class TestDelegate
    {
        static int num = 10;
        public static int AddNum(int p)
        {
            num += p;
            return num;
        }
        public static int MultNum(int q)
        {
            num *= q;
            return num;
        }
        public static int getNum()
        {
            return num;
        }
        static void Main(string[] args)
        {
            NumberChanger nc1 = new NumberChanger(AddNum);
            NumberChanger nc2 = new NumberChanger(MultNum);
            nc1(25);
            Console.WriteLine("Value of Num: {0}", getNum());
            nc2(5);
            Console.WriteLine("Value of Num: {0}", getNum());
        }
    }
}
```

## C# & .NET FRAMEWORK LAB MANUAL

**Write a C# programs to demonstrate the concepts of multi cast Delegates.**

```
using System;
delegate void mdelegate();
namespace DelegateAppl
{
    class dm
    {
        static public void display()
        {
            Console.WriteLine("new delhi");
        }
        static public void print()
        {
            Console.WriteLine("new york");
        }
    }
    class mtest
    {
        public static void Main()
        {
            mdelegate m1=new mdelegate(dm.display);
            mdelegate m2=new mdelegate(dm.print);
            mdelegate m3=m1+m2;
            mdelegate m4=m2+m1;
            mdelegate m5=m3-m2;
            m3();
            m4();
            m5();
        }
    }
}
```

# C# & .NET FRAMEWORK LAB MANUAL

## LAB-5:

### 1. Write a C# programs to demonstrate the concepts of Label, Text Box and Button controls.

```
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 WindowsFormsApplication5
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void textBox1_Validating(object sender, CalEventArgs e)
        {
            TextBox tb = sender as TextBox;
            if (tb.Name == "textBox1")
            {
                if (tb.Text.Trim().Length == 0)
                {
                    MessageBox.Show("cant leave empty");
                    e.Cancel = true;
                    return;
                }
            }
        }
    }
}
```



## C# & .NET FRAMEWORK LAB MANUAL

```
private void textBox2_Validating(object sender, CancelEventArgs e)
{
    TextBox tb = sender as TextBox;
    if (tb.Name == "textBox2")
    {
        if (tb.Text.Trim().Length == 0)
        {
            MessageBox.Show("cant leave empty");
            e.Cancel = true;
            return;
        }
    }
}

private void button1_Click(object sender, EventArgs e)
{
    string temp;
    temp = textBox1.Text;
    textBox1.Text = textBox2.Text;
    textBox2.Text = temp;
    button1.Visible = false;
}

private void textBox2_Leave(object sender, EventArgs e)
{
    TextBox tb = sender as TextBox;
    if (tb.Name == "TextBox2")
    {
        if (tb.Text.Trim().Length == 0)
        {
            MessageBox.Show("cant leave empty");
            return;
        }
    }
}
```

## C# & .NET FRAMEWORK LAB MANUAL

```
private void textBox1_Leave(object sender, EventArgs e)
{
    TextBox tb = sender as TextBox;
    if (tb.Name == "TextBox1")
    {
        if (tb.Text.Trim().Length == 0)
        {
            MessageBox.Show("cant leave empty");
            return;
        }
    }
}
```

# C# & .NET FRAMEWORK LAB MANUAL

## 2. Create a windows application in c# to demonstrate comboBox and ListBox controls using button click event.

```
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 WindowsFormsApplication22
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            MessageBox.Show(comboBox1.SelectedItem.ToString());
            string str = " ";
            foreach (object ob in listBox1.SelectedItems)
            {
                str += ob.ToString() + " ,";
            }
            str = str.Substring(0, str.Length);
            MessageBox.Show(str);
        }
    }
}
```

# C# & .NET FRAMEWORK LAB MANUAL

## Lab-6:

1. Create a Windows application in C# for registration form and fill the details and when you click the submit button it display the details in the message box.

### Properties:

1. Textbox2->passwordchar=\*,maxlength=10

2. Textbob3->passwordchar=\*

3. Textbox5->multiline=true

```
using System;
using System.Windows.Forms;
namespace WindowsFormsApplication21
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void textBox1_Validating(object sender, CancelEventArgs e)
        {
            TextBox tb = sender as TextBox;
            if (tb.Name == "textBox1")
            {
                if (tb.Text.Trim().Length == 0)
                {
                    MessageBox.Show("cannot leave empty");
                    e.Cancel = true;
                    return;
                }
            }
        }
    }
}
```

## C# & .NET FRAMEWORK LAB MANUAL

```
private void textBox2_Validating(object sender, CancelEventArgs e)
{
    TextBox tb = sender as TextBox;
    if (tb.Name == "textBox2")
    {
        if (tb.Text.Trim().Length == 0)
        {
            MessageBox.Show("cannot leave empty");
            e.Cancel = true;
            return;
        }
    }
}

private void textBox3_Validating(object sender, CancelEventArgs e)
{
    TextBox tb = sender as TextBox;
    if (tb.Name == "textBox3")
    {
        if (textBox2.Text.Trim() != textBox3.Text.Trim())
        {
            MessageBox.Show("not matched");
            e.Cancel = true;
            return;
        }
    }
}

private void textBox4_KeyPress(object sender, KeyPressEventArgs e)
{
    if (char.IsDigit(e.KeyChar) == false)
    {
        MessageBox.Show("enter only numbers");
        e.Handled = true;
        return;
    }
}
```

## C# & .NET FRAMEWORK LAB MANUAL

```
    }  
}  
private void button1_Click(object sender, EventArgs e)  
{  
    MessageBox.Show("name=" + textBox1.Text + "\n password=" +  
        textBox2.Text + "\n confirm password=" + textBox3.Text +  
        "\n age=" + textBox4.Text + "\n address=" +  
        textBox5.Text);  
}  
}
```

# C# & .NET FRAMEWORK LAB MANUAL

**2. Create a Windows application in C# having two text boxes and three buttons named as factorial, prime, factorial series. When you click any button the resultant value will be displayed on the second textbox.**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Text;
using System.Windows.Forms;
namespace WindowsFormsApplication11
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            int n, i = 1, fact = 1;
            n=int.Parse(textBox1.Text);
            while (i <= n)
            {
                fact = fact * i;
                i++;
            }
            textBox2.Text = fact.ToString();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            int i, j, c;
            int n = int.Parse(textBox1.Text);
            textBox2.Text = "";
        }
    }
}
```

## C# & .NET FRAMEWORK LAB MANUAL

```
        for (j = 2; j <= n; j++)
        {
            c = 0; i = 1;
            while (i <= j)
            {
                if (j % i == 0)
                {
                    c++;
                }
                i++;
            }
            if (c <= 2)
            {
                textBox2.Text = textBox2.Text + j.ToString() + ",";
            }
        }
    }

    private void button3_Click(object sender, EventArgs e)
    {
        int prev=0, cur=1, next;
        int n = int.Parse(textBox1.Text);
        textBox2.Text = prev.ToString() + cur.ToString();
        next = prev + cur;
        while (next <=n)
        {
            textBox2.Text = textBox2.Text + next.ToString();
            prev = cur;
            cur = next;
            next = prev + cur;
        }
    }
}
```



# C# & .NET FRAMEWORK LAB MANUAL

## LAB-7:

### 1. Create a ADO.NET application in C# to verify if the connection is established with OLEDB and MS-ACCESS.

```
using System;
using System.Windows.Forms;
using System.Data.Odbc;
namespace WindowsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            string connectionString = null; OdbcConnection cnn ;
            connectionString = "Driver={Microsoft Access Driver
                (*.mdb)};DBQ=yourdatabasename.mdb;";
            cnn = new OdbcConnection(connectionString);
            try
            {
                cnn.Open();
                MessageBox.Show ("Connection Open ! ");
                cnn.Close();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Can not open connection ! ");
            }
        }
    }
}
```

# C# & .NET FRAMEWORK LAB MANUAL

## 2. Create a ADO.NET applications in C# to demonstrate the Data Reader, Data Set, Data Adapter and Data View Objects.

```
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;
using System.Data.OleDb;
namespace datareaderado.net
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        static OleDbConnection con = new
            OleDbConnection("provider=msdaora.1;user
            id=system;password=uday");
        OleDbDataReader dr;
        Label[] L;
        int x = 50;
        TextBox[] T;
        private void Form1_Load(object sender, EventArgs e)
        {
            OleDbCommand c1 = new OleDbCommand("select * from tab", con);
            con.Open();
            OleDbDataReader d = c1.ExecuteReader();
            while (d.Read() == true)
            {
```

## C# & .NET FRAMEWORK LAB MANUAL

```
        comboBox1.Items.Add(d[0].ToString());
    }
    con.Close();
}
private void button1_Click(object sender, EventArgs e)
{
    if (comboBox1.SelectedIndex == -1)
    {
        MessageBox.Show("kindly select a table first");
    }
    else
    {
        con.Open();
        OleDbCommand cmd = new OleDbCommand("select * from
            comboBox1.SelectedIndex.ToString()", con);
        dr = cmd.ExecuteReader();
        MessageBox.Show("no of cols" + dr.FieldCount);
        L = new Label[dr.FieldCount];
        T = new TextBox[dr.FieldCount];
        for (int i = 0; i < dr.FieldCount; i++)
        {
            L[i] = new Label();
            L[i].Text = dr.GetName(i);
            L[i].Location=new Point(100,x);
            this.Controls.Add(L[i]);
            T[i] = new TextBox();
            T[i].Location = new Point(200, x);
            this.Controls.Add(T[i]);
            x = x + 30;
        }
    }
}
```

## C# & .NET FRAMEWORK LAB MANUAL

```
private void button2_Click(object sender, EventArgs e)
{
    if (dr.Read() == true)
    {
        for (int i = 0; i < dr.FieldCount; i++)
            T[i].Text = dr[i].ToString();
    }
    else
        MessageBox.Show("No more records");
}

private void button3_Click(object sender, EventArgs e)
{
    for (int i = 0; i < dr.FieldCount; i++)
    {
        this.Controls.Remove(L[i]);
        this.Controls.Remove(T[i]);
    }
    con.Close();
    x = 50;
}
}
```

# C# & .NET FRAMEWORK LAB MANUAL

```
OleDbConnection ocon;
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;
using System.Data.OleDb;
namespace WindowsFormsApplication25
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        OleDbConnection ocon;
        private void button1_Click(object sender, EventArgs e)
        {
            ocon=new OleDbConnection("Provider=MSDAORA.1;user
                id=system; password=raja;");
            ocon.Open();
            MessageBox.Show(ocon.State.ToString());
            ocon.Close();
            MessageBox.Show(ocon.State.ToString());
        }
    }
}
```

# C# & .NET FRAMEWORK LAB MANUAL

**Create a ADO.NET application in C#, to create a table and insert values into created table.**

```
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;
using System.Data.OleDb;
namespace WindowsFormsApplication25
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        OleDbConnection ocon;
        OleDbCommand cmd1,cmd2;
        private void button1_Click(object sender, EventArgs e)
        {
            con =new OleDbConnection("Provider=MSDAORA.1;
            user id=system;password=raja;");
            ocon.Open();
            cmd1=new OleDbCommand("create table
            student1(sid number,sname
            varchar(10))",ocon);
            cmd2=new OleDbCommand("insert into student1
            values('1','raja')",ocon);
            cmd1.ExecuteNonQuery();
            MessageBox.Show("table created");
        }
    }
}
```

## C# & .NET FRAMEWORK LAB MANUAL

```
        cmd2.ExecuteNonQuery();  
        MessageBox.Show("1 row inserted");  
    }  
}  
}
```

## C# & .NET FRAMEWORK LAB MANUAL

**Create an ADO.NET application in C#, to retrieve the values from the table using DataReader object.**

```
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;
using System.Data.OleDb;
namespace WindowsFormsApplication25
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        OleDbConnection ocon;
        OleDbDataReader dr;
        private void Form1_Load(object sender, EventArgs e)
        {
            ocon = new OleDbConnection("Provider=MSDAORA.1;
                                     user id=system; password=raja");
            ocon.Open();
            OleDbCommand cmd2 = new OleDbCommand("select * from
                                                  student4", ocon);

            dr = cmd2.ExecuteReader();
            label1.Text = dr.GetName(0);
            label2.Text = dr.GetName(1);
        }
    }
}
```



## C# & .NET FRAMEWORK LAB MANUAL

```
private void button1_Click(object sender, EventArgs e)
{
    if (dr.Read())
    {
        textBox1.Text = dr.GetValue(0).ToString();
        textBox2.Text = dr.GetValue(1).ToString();
    }
    else
    {
        MessageBox.Show("no more records");
    }
}
private void button2_Click(object sender, EventArgs e)
{
    ocon.Close();
    MessageBox.Show(ocon.State.ToString());
}
}
```

# C# & .NET FRAMEWORK LAB MANUAL

## Create an ADO.NET application in C#, to demonstrate DataAdapter object

```
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;
using System.Data.OleDb;
namespace WindowsFormsApplication25
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        OleDbConnection ocon;
        OleDbDataReader dr;
        DataSet ds = new DataSet();
        OleDbDataAdapter da;
        int i, c, n;
        private void Form1_Load(object sender, EventArgs e)
        {
            ocon = new OleDbConnection("Provider=MSDAORA.1;
            user id=system; password=raja");
            OleDbCommand cmd = new OleDbCommand("select *
            from student4", ocon);
            OleDbDataAdapter da = new OleDbDataAdapter("select *
            from student4", ocon);
            ocon.Open();
```

## C# & .NET FRAMEWORK LAB MANUAL

```
        dr = cmd.ExecuteReader();
        label1.Text = dr.GetName(0);
        label2.Text = dr.GetName(1);
        da.Fill(ds, "student4");
        while (dr.Read() != false)
        {
            c++;
        }
        n = c-1;
    }
    private void button1_Click(object sender, EventArgs e)
    {
        i = 0;
        textBox1.Text = ds.Tables["student4"].Rows[i][0].ToString();
        textBox2.Text = ds.Tables["student4"].Rows[i][1].ToString();
    }
    private void button2_Click(object sender, EventArgs e)
    {
        if (i < n)
        {
            i++;
            textBox1.Text = ds.Tables["student4"].Rows[i][0].ToString();
            textBox2.Text = ds.Tables["student4"].Rows[i][1].ToString();
        }
        else
        {
            MessageBox.Show("no more records");
        }
    }
    private void button3_Click(object sender, EventArgs e)
    {
        if (i > 0)
        {
```

## C# & .NET FRAMEWORK LAB MANUAL

```
        i--;
        textBox1.Text = ds.Tables["student4"].Rows[i][0].ToString();
        textBox2.Text = ds.Tables["student4"].Rows[i][1].ToString();
    }
    else
    {
        MessageBox.Show("no more records");
    }
}
private void button4_Click(object sender, EventArgs e)
{
    i = n;
    textBox1.Text = ds.Tables["student4"].Rows[i][0].ToString();
    textBox2.Text = ds.Tables["student4"].Rows[i][1].ToString();
}
}
```

# C# & .NET FRAMEWORK LAB MANUAL

**Create an ADO.NET application in C#, to demonstrate dataGridView Control.**

```
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;
using System.Data.OleDb;
namespace WindowsFormsApplication27
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        DataSet dset=new DataSet();
        OleDbConnection ocon;
        private void Form1_Load(object sender, EventArgs e)
        {
            ocon = new OleDbConnection("Provider=MSDAORA.1;user
                id=system; password=raja");
            OleDbDataAdapter ad = new OleDbDataAdapter(" select *
                from student4", ocon);
            ad.Fill(dset,"student4");
            dataGridView1.DataSource = dset.Tables["student4"];
        }
    }
}
```

# C# & .NET FRAMEWORK LAB MANUAL

## LAB-8

### 1. Develop the Static Webpages using HTML and some validations along with Java Script.

```
<html>
  <head>
    <script language="javascript">
      function f1( )
      {
        D=new Date( );
        h=d.getHours( );
        m=d.getMinutes ( );
        s=d.getSeconds( );
        F.t1.Value=h+":"+m+":"+s;
        Window.SetTimeout("F1()",1000)
      }
    </script>
  </head>
  <body onload="F1( )">
    <form name="f">
      Time:<input type="text" name="t1">
    </form>
  </body>
</html>
```

# C# & .NET FRAMEWORK LAB MANUAL

## LAB-9

### 1. Design an ASP.NET Webpage to work with Cross page and Post back Submissions.

#### Creating a website to work with crosspage submission.

- Select ASP.NET empty website.
- Add web form and name it as login.aspx
- Goto design view of login.aspx

The image shows a design view of a login form. It consists of a rounded rectangular container with a thin black border. Inside the container, there are two text input fields. The first field is labeled 'Uname:' and the second is labeled 'Pwd :'. Below the container, there is a rectangular button labeled 'Submit'.

- Place two labels, two textboxes and one submit button
- **Label1 properties:**
  - id=l1
  - Text=username
- **Label2 properties:**
  - id=l2
  - Text=password
- **textBox1 properties:**
  - id=t1
- **textBox2 properties:**
  - id=t2
  - textmode=password
- **button1 properties:**
  - id=b1
  - text=submit
  - postbackurl=validate.aspx
  - add webform(place one more web form into asp.net website) and name it as validate.aspx

## C# & .NET FRAMEWORK LAB MANUAL

- goto design view of validate.aspx
- place one label
- lebel1 properties
- id=lb
- code for page load event

```
{  
    String uname=Request.Form["t1"];  
    String pwd=Request.Form["t2"];  
    lb.Text="wel come to"+uname;  
}
```

- to set login page as starting page
- goto solution explorer=>right click on login.aspx=>set as start page
- ctrl+f5



# C# & .NET FRAMEWORK LAB MANUAL

## Creating a website to work with postback submission

- Select ASP.NET empty website.
- Add web form and name it as customer.aspx
- Go to design view of customer.aspx

ACCNO:

NAME :

Display

- textBox2 properties:
  - visible=false
- textBox3 properties:
  - visible=false
- button1\_click event logic

```
{  
    String[,]={{"60","uday","5000"},"{"50","raja1","10000"}};  
    bool b=false;  
    if(s[I,0]==textBox1.Text)  
    {  
        textBox2.Text=s[i,1];  
        textBox3.Text=s[I,2];  
        textBox2.Visible=true;  
        textBox3.Visible=true;  
        b=true;  
        break;  
    }  
    if(b==false)  
        Response.Write("accno not existing");  
}
```

- ctrl+f5

# C# & .NET FRAMEWORK LAB MANUAL

## 2. Design an ASP.NET Webpage to work with Dropdown list and ListBox controls.

### Procedure:

1. Select ASP.NET empty website .
2. Add web form
3. Place one dropdownlist, two listboxes and four button on default.aspx design view.

### Properties:

#### Dropdownlist1 properties:

- autopostback=true
- items= Microsoft, Microsoft1, Microsoft1

#### ListBox1 properties:

- selectMode=multiple

#### ListBox2 properties:

- selectMode=multiple

#### Button1 properties:

- Text =one(>)

#### Button2 properties:

- Text=multiple(>>)

#### Button3 properties:

- Text=removeone

#### Button4 properties:

- Text=removeall

```
using System;
```

```
using System.Web;
```

```
using System.Web.UI;
```

```
using System.Web.UI.WebControls;
```

```
namespace WebApplication2
```

```
{
```

```
    public partial class WebForm1 : System.Web.UI.Page
```

```
    {
```

```
        protected void DropDownList1_SelectedIndexChanged(object sender,  
            EventArgs e)
```

## C# & .NET FRAMEWORK LAB MANUAL

```
{
    ListBox1.Items.Clear();
    switch (DropDownList1.SelectedValue)
    {
        case "microsoft": ListBox1.Items.Add("10");
                          ListBox1.Items.Add("20");
                          break;
        case " microsoft1": ListBox1.Items.Add("100");
                           ListBox1.Items.Add("200");
                           break;
        case " microsoft2":ListBox1.Items.Add("1000");
                           ListBox1.Items.Add("2000");
                           break;
    }
}
protected void Button1_Click(object sender, EventArgs e)
{
    ListBox2.Items.Add(ListBox1.SelectedItem);
}
protected void Button2_Click(object sender, EventArgs e)
{
    for (byte i = 0; i < ListBox1.Items.Count; i++)
        ListBox2.Items.Add(ListBox1.Items[i]);
}
protected void Button3_Click(object sender, EventArgs e)
{
    ListBox2.Items.RemoveAt(ListBox2.SelectedIndex);
}
protected void Button4_Click(object sender, EventArgs e)
{
    ListBox2.Items.Clear();
}
}
```

# C# & .NET FRAMEWORK LAB MANUAL

## LAB-10

### 1. Develop a Registration Form with all Validation Controls.

#### Procedure:

1. Select ASP.NET empty website and name it as registration
2. Add webform(default.aspx)
3. Place 5 labels, 5 textboxes and 1 button
4. Place one RequiredFieldValidator,  
one RangeValidator,  
one CompareValidator,  
one RegularExpressionValidator and  
one ValidationSummary control on default.aspx design view.

#### Properties:

##### textBox2:

- text=password

##### textBox3:

- text=password

##### RequiredFieldValidator1:

- ControlToValidate=textBox1
- ErrorMessage=Enter something

##### RangeValidator1:

- ControlToValidate=textBox4
- ErrorMessage=should be greater than 18
- Type=integer

##### CompareValidator1:

- ControlToCompare=textBox2
- ControlToValidate=textBox3
- ErrorMessage=doesnot match password

##### RegularExpressionValidator1:

- ControlToValidate=textBox5
- ErrorMessage=invalid email id
- validationExpression=select internet email address

##### ValidationSummary:

- HeaderText=you have received the following errors.

# C# & .NET FRAMEWORK LAB MANUAL

## 2. Create a Web Service for all Arithmetic operations.

### Procedure:

1. Go to visual studio File==>new==>website
2. In solution explorer Right click on project path and select webservice and click on add.
3. Define web methods

```
[WebMethod]
public int add(int a, int b)
{
    return a+b;
}

[WebMethod]
public int sub(int a, int b)
{
    return a-b;
}

[WebMethod]
public int mul(int a, int b)
{
    return a*b;
}

[WebMethod]
public int div(int a, int b)
{
    return a/b;
}

[WebMethod]
public int moddiv(int a, int b)
{
    return a%b;
}
```

- After writing the logic then go to debug select start without debugging option
- The output window will be displayed in WSDL XML document

# C# & .NET FRAMEWORK LAB MANUAL

## LAB-11

### 1. Write a C# program to implement Private Assemblies.

#### Procedure:

- Open a new class library template(name as raja)

using System;

namespace raja

{

    public class pa1

    {

        public string m1()

        {

            return "pa1 test";

        }

    }

    namespace sekhar

    {

        public class pa2

        {

            public string m2()

            {

                return "pa2 test";

            }

        }

    }

}

- Now build the project, it will generate .dll file in their project folder.

#### CONSUMING THE PRIVATE ASSEMBLY:

- **Open winform app project(name as privateassembly)**

- **Place abutton**

- **To Add assembly to private assembly**

Right click on Project path ==>add new item==>add reference==>browse==>select raja.dll file

- Then write the code under button click event

## C# & .NET FRAMEWORK LAB MANUAL

```
using raja;
using raja.sekhar;
namespace WindowsFormsApplication32
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            pa1 p1 = new pa1();
            MessageBox.Show(p1.m1());
            pa2 p2 = new pa2();
            MessageBox.Show(p2.m2());
        }
    }
}
```

# C# & .NET FRAMEWORK LAB MANUAL

**Write a C# program to implement shared Assemblies.**

**Procedure:**

- Open class library template(name as uday)

```
using System;
namespace uday
{
    public class r
    {
        public string a()
        {
            return "this is uday's assembly";
        }
    }
}
```

- To create a strong name go to visual studio command prompt and set the path upto debug folder(uday project path).

```
sn -k key.snk
```

- To link key.snk to uday

Right click on Project path ==> properties==> signing==>click checkbox==>select browse from combobox==>choose key.snk.

- Now build the project

```
build==>build uday
```

- Register raja.dll with GAC(global assembly cache)(the location of GAC folder is c:\windows\microsoft.net\assembly\GAC\_MSIL)

```
gacutil -i uday.dll
```

**CONSUMING SHARED ASSEMBLY:**

- **Open winform project**
- **Place a button**
- **Right click on Project path ==>add reference==>browse==>uday.dll**

```
using uday;
namespace WindowsFormsApplication33
{
```



## C# & .NET FRAMEWORK LAB MANUAL

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
    private void button1_Click(object sender, EventArgs e)
    {
        r b = new r();
        MessageBox.Show(b.a());
    }
}
}
```

# C# & .NET FRAMEWORK LAB MANUAL

## 2. Write a C# program to implement Multi Threading using System;

using System.Threading;

```
namespace ConsoleApplication11
```

```
{
```

```
    class Program
```

```
    {
```

```
        public static void m1()
```

```
        {
```

```
            for (int i = 0; i <= 10; i++)
```

```
            {
```

```
                Console.WriteLine(i + "");
```

```
                Thread.Sleep(500);
```

```
            }
```

```
        }
```

```
        public static void m2()
```

```
        {
```

```
            for (int k = 11; k <= 20; k++)
```

```
            {
```

```
                Console.WriteLine(k + "");    Thread.Sleep(500);
```

```
            }
```

```
        }
```

```
    }
```

```
    class test
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            ThreadStart ts1 = new ThreadStart(Program.m1);
```

```
            ThreadStart ts2 = new ThreadStart(Program.m2);
```

```
            Thread t1 = new Thread(ts1);  Thread t2 = new Thread(ts2);
```

```
            t1.Start();  t2.Start();
```

```
        }
```

```
    }
```

```
}
```

# C# & .NET FRAMEWORK LAB MANUAL

## Write a C# program to implement Thread Synchronization

```
using System;
using System.Threading;
namespace ConsoleApplication13
{
    class Program
    {
        Mutex m = new Mutex();
        public void m1()
        {
            m.WaitOne();
            for (int i = 1; i <= 10; i++)
            {
                Console.WriteLine(i + "");
                Thread.Sleep(500);
            }
            m.ReleaseMutex();
        }
    }
    class test
    {
        static void Main(string[] args)
        {
            Program p = new Program();
            ThreadStart ts1 = new ThreadStart(p.m1);
            Thread t1= new Thread(ts1);
            Thread t2 = new Thread(ts1);
            t1.Start(); t2.Start();
        }
    }
}
```



**RAJEEV GANDHI MEMORIAL COLLEGE OF ENGINEERING & TECHNOLOGY**

**(AUTONOMOUS)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

---

### **Evaluation Procedure for Internal Laboratory Examinations:**

1. Of the 25 marks for internal, 10 marks will be awarded for day-to-day work and 10 marks to be awarded for the Record work and 5 marks to be awarded by conducting an internal laboratory test.
2. Concerned Teachers have to do necessary corrections with explanations.
3. Concerned Lab teachers should enter marks in index page.
4. Internal exam will be conducted by two Staff members.

Dr.K. Subba Reddy

Professor & Head Dept. of CSE.



**RAJEEV GANDHI MEMORIAL COLLEGE OF ENGINEERING & TECHNOLOGY**

**(AUTONOMOUS)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

---

### **Evaluation Procedure for External Laboratory Examinations:**

1. For Practical subjects there is a continuous evaluation during the semester for 25 Sessional marks and 50 end examination marks.
2. The end examination shall be conducted by the teacher concerned (Internal Examiner) and another External Examiner, recommended by Head of the Department with the approval of principal.

Evaluation procedure for external lab examination:

1. Procedure for the program	----- 20M
2. Execution of the program	----- 15M
3. Viva voce	----- 15M
	-----
Total	50M
	-----

Dr.K. Subba Reddy

Professor & Head Dept. of CSE.