

Static and non-static methods in C#:

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

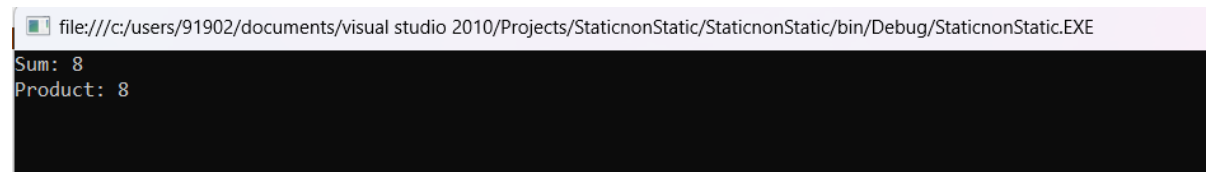
namespace StaticnonStatic
{
    class MathHelper
    {
        public static int Add(int a, int b)
        {
            return a + b;
        }

        public int Multiply(int a, int b)
        {
            return a * b;
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            // Static method example
            int sum = MathHelper.Add(5, 3);
            Console.WriteLine("Sum: " + sum);

            // Non-static method example
            MathHelper mathHelper = new MathHelper();
            int product = mathHelper.Multiply(4, 2);
            Console.WriteLine("Product: " + product);

            Console.ReadLine();
        }
    }
}
```



```
file:///c:/users/91902/documents/visual studio 2010/Projects/StaticnonStatic/StaticnonStatic/bin/Debug/StaticnonStatic.EXE
Sum: 8
Product: 8
```


Extension methods in C#:

using System;

```
static class StringExtensions
{
    public static string Reverse(this string str)
    {
        char[] charArray = str.ToCharArray();
        Array.Reverse(charArray);
        return new string(charArray);
    }
}
```

```
class Program
{
    static void Main(string[] args)
    {
        string text = "Hello, World!";
        string reversed = text.Reverse();
        Console.WriteLine(reversed);

        Console.ReadLine();
    }
}
```

 file:///c:/users/91902/documents/visual studio 2010/Projects/extensionmethod/extensionmethod/bin/Debug/extensionmethod.EXE

dlroW ,olleH

Interface inheritance in C#:

```
interface IShape
```

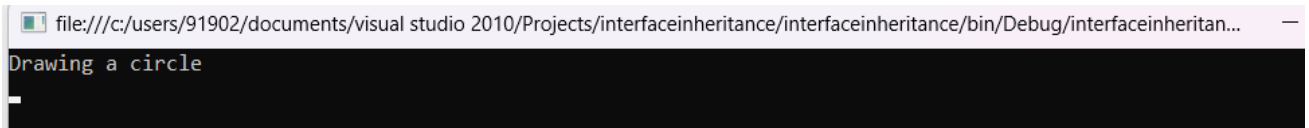
```
{  
    void Draw();  
}
```

```
class Circle : IShape
```

```
{  
    public void Draw()  
    {  
        Console.WriteLine("Drawing a circle");  
    }  
}
```

```
class Program
```

```
{  
    static void Main(string[] args)  
    {  
        Circle circle = new Circle();  
        circle.Draw();  
  
        Console.ReadLine();  
    }  
}
```



Single inheritance in C#:

```
class Vehicle
```

```
{  
    public void Drive()  
    {  
        Console.WriteLine("Driving the vehicle");  
    }  
}
```

```
class Car : Vehicle
```

```
{  
    public void Accelerate()  
    {  
        Console.WriteLine("Accelerating the car");  
    }  
}
```

```
class Program
```

```
{  
    static void Main(string[] args)  
    {  
        Car car = new Car();  
        car.Drive();  
        car.Accelerate();  
  
        Console.ReadLine();  
    }  
}
```

file:///c:/users/91902/documents/visual studio 2010/Projects/singleinheritance/singleinheritance/bin/Debug/singleinheritance.EXE

```
Driving the vehicle  
Accelerating the car  
-
```

Multiple inheritance in C#:

```
interface IShape
```

```
{  
    void Draw();  
}
```

```
interface IMovable
```

```
{  
    void Move();  
}
```

```
class Rectangle : IShape, IMovable
```

```
{  
    public void Draw()  
    {  
        Console.WriteLine("Drawing a rectangle");  
    }  
  
    public void Move()  
    {  
        Console.WriteLine("Moving the rectangle");  
    }  
}
```

```
class Program
```

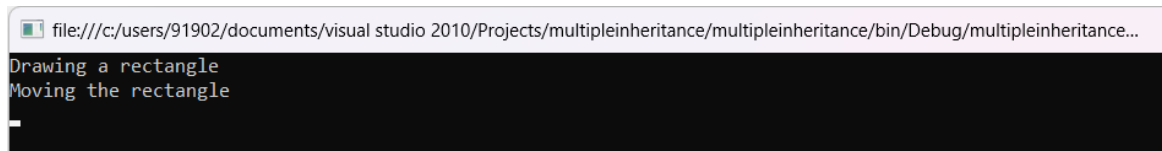
```
{
```

```

static void Main(string[] args)
{
    Rectangle rectangle = new Rectangle();
    rectangle.Draw();
    rectangle.Move();

    Console.ReadLine();
}
}

```



Hybrid inheritance in C#:

class Animal

```

{
    public void Eat()
    {
        Console.WriteLine("Animal is eating");
    }
}

```

interface ISwim

```

{
    void Swim();
}

```

interface IFly

```

{

```

```

        void Fly();
    }

class Dolphin : Animal, ISwim
{
    public void Swim()
    {
        Console.WriteLine("Dolphin is swimming");
    }
}

class Bird : Animal, IFly
{
    public void Fly()
    {
        Console.WriteLine("Bird is flying");
    }
}

class FlyingFish : Animal, ISwim, IFly
{
    public void Swim()
    {
        Console.WriteLine("Flying fish is swimming");
    }

    public void Fly()
    {
        Console.WriteLine("Flying fish is flying");
    }
}

```

```
}
```

```
class Program
```

```
{
```

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

```
    {
```

```
        Dolphin dolphin = new Dolphin();
```

```
        dolphin.Eat();
```

```
        dolphin.Swim();
```

```
        Bird bird = new Bird();
```

```
        bird.Eat();
```

```
        bird.Fly();
```

```
        FlyingFish flyingFish = new FlyingFish();
```

```
        flyingFish.Eat();
```

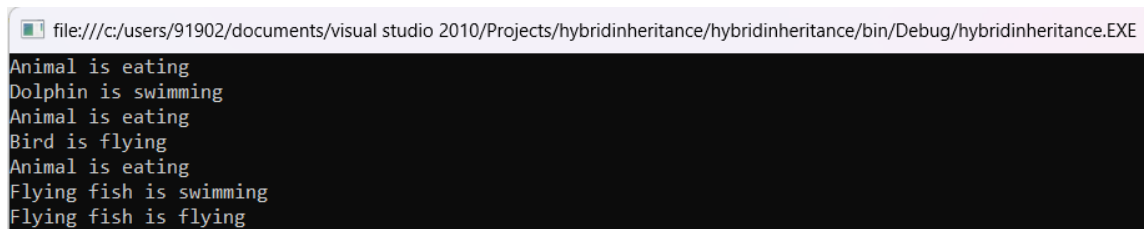
```
        flyingFish.Swim();
```

```
        flyingFish.Fly();
```

```
        Console.ReadLine();
```

```
    }
```

```
}
```



```
file:///c:/users/91902/documents/visual studio 2010/Projects/hybridinheritance/hybridinheritance/bin/Debug/hybridinheritance.EXE
Animal is eating
Dolphin is swimming
Animal is eating
Bird is flying
Animal is eating
Flying fish is swimming
Flying fish is flying
```


Encapsulation in C#:

```
class Person
{
    private string name;
    private int age;
    public string Name
    {
        get { return name; }
        set { name = value; }
    }
    public int Age
    {
        get { return age; }
        set { age = value; }
    }
}

class Program
{
    static void Main(string[] args)
    {
        Person person = new Person();
        person.Name = "John";
        person.Age = 30;

        Console.WriteLine("Name: " + person.Name);
        Console.WriteLine("Age: " + person.Age);
        Console.ReadLine();
    }
}
```

file:///c:/users/91902/documents/visual studio 2010/Projects/encapsulation/encapsulation/bin/Debug/encapsulation.EXE

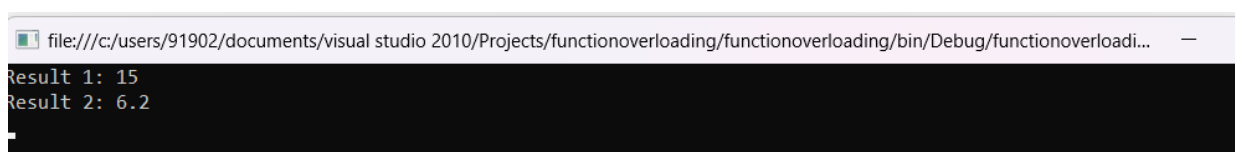
Name: Harshad
Age: 20

Function overloading in C#:

```
class Calculator
{
public int Add(int num1, int num2)
    {
        return num1 + num2;
    }
public double Add(double num1, double num2)
    {
        return num1 + num2;
    }
}
class Program
{
    static void Main(string[] args)
    {
        Calculator calculator = new Calculator();
        int result1 = calculator.Add(5, 10);
        double result2 = calculator.Add(2.5, 3.7);

        Console.WriteLine("Result 1: " + result1);
        Console.WriteLine("Result 2: " + result2);

        Console.ReadLine();
    }
}
```

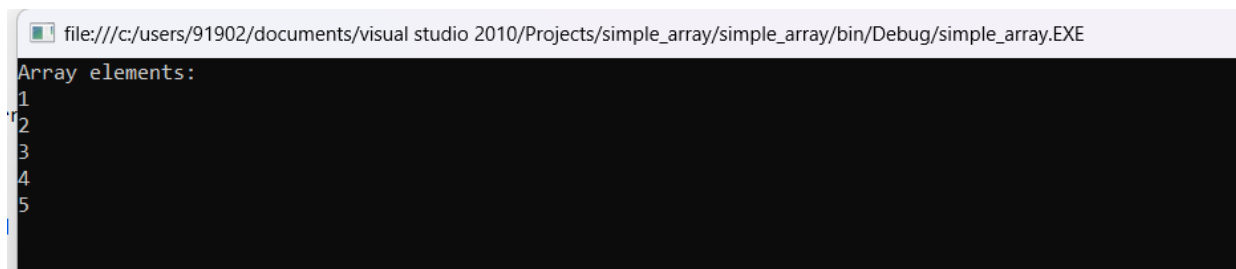


```
file:///c:/users/91902/documents/visual studio 2010/Projects/functionoverloading/functionoverloading/bin/Debug/functionoverloadi...
Result 1: 15
Result 2: 6.2
```

Arrays in C#:

class Program

```
{  
    static void Main(string[] args)  
    {  
        int[] numbers = new int[5] { 1, 2, 3, 4, 5 };  
  
        Console.WriteLine("Array elements:");  
        for (int i = 0; i < numbers.Length; i++)  
        {  
            Console.WriteLine(numbers[i]);  
        }  
  
        Console.ReadLine();  
    }  
}
```



```
file:///c:/users/91902/documents/visual studio 2010/Projects/simple_array/simple_array/bin/Debug/simple_array.EXE  
Array elements:  
1  
2  
3  
4  
5
```

Multidimensional arrays in C#:

class Program

```
{  
    static void Main(string[] args)  
    {  
        int[,] matrix = new int[3, 3]  
        {  
            { 1, 2, 3 },  
            { 4, 5, 6 },  
            { 7, 8, 9 }  
        }  
    }  
}
```

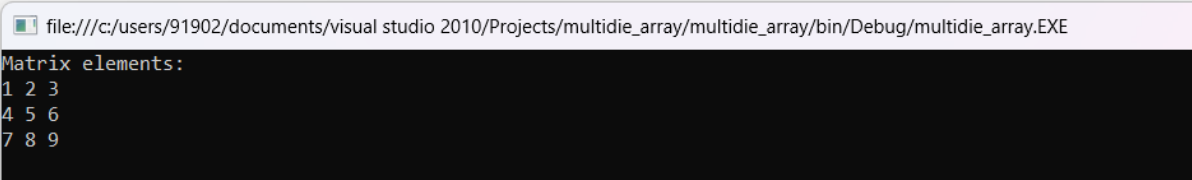
```

        { 4, 5, 6 },
        { 7, 8, 9 }
    };

    Console.WriteLine("Matrix elements:");
    for (int row = 0; row < 3; row++)
    {
        for (int col = 0; col < 3; col++)
        {
            Console.Write(matrix[row, col] + " ");
        }
        Console.WriteLine();
    }

    Console.ReadLine();
}

```



```

file:///c:/users/91902/documents/visual studio 2010/Projects/multidie_array/multidie_array/bin/Debug/multidie_array.EXE
Matrix elements:
1 2 3
4 5 6
7 8 9

```

Jagged arrays in C#:

```


class Program
{
    static void Main(string[] args)
    {
        int[][] jaggedArray = new int[3][];
        jaggedArray[0] = new int[] { 1, 2, 3 };
        jaggedArray[1] = new int[] { 4, 5 };
    }
}

```

```
jaggedArray[2] = new int[] { 6, 7, 8, 9 };

Console.WriteLine("Jagged array elements:");
for (int i = 0; i < jaggedArray.Length; i++)
{
    for (int j = 0; j < jaggedArray[i].Length; j++)
    {
        Console.Write(jaggedArray[i][j] + " ");
    }
    Console.WriteLine();
}

Console.ReadLine();
}
```



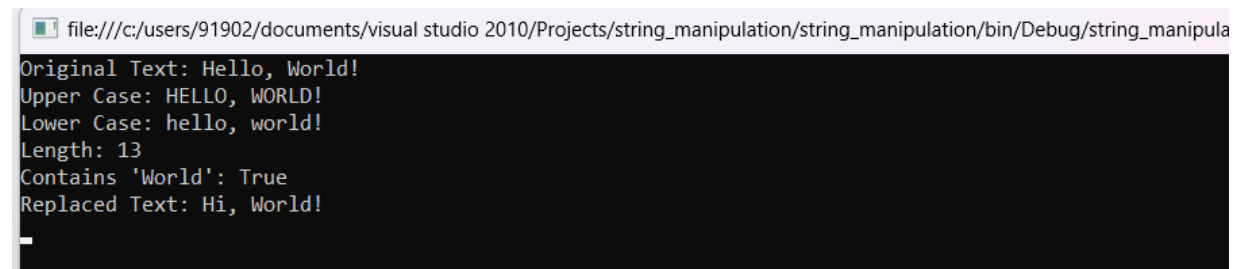
The screenshot shows a console window with the title bar "file:///c:/users/91902/documents/visual studio 2010/Projects/jagged_array/jagged_array/bin/Debug/jagged_array.EXE". The output text is as follows:

```
Jagged array elements:
1 2 3
4 5
6 7 8 9
```

String manipulation using string in C#:

class Program

```
{  
    static void Main(string[] args)  
    {  
        string text = "Hello, World!";  
        string upperCase = text.ToUpper();  
        string lowerCase = text.ToLower();  
        int length = text.Length;  
        bool containsWorld = text.Contains("World");  
        string replaced = text.Replace("Hello", "Hi");  
  
        Console.WriteLine("Original Text: " + text);  
        Console.WriteLine("Upper Case: " + upperCase);  
        Console.WriteLine("Lower Case: " + lowerCase);  
        Console.WriteLine("Length: " + length);  
        Console.WriteLine("Contains 'World': " + containsWorld);  
        Console.WriteLine("Replaced Text: " + replaced);  
  
        Console.ReadLine();  
    }  
}
```



file:///c:/users/91902/documents/visual studio 2010/Projects/string_manipulation/string_manipulation/bin/Debug/string_manipula

```
Original Text: Hello, World!  
Upper Case: HELLO, WORLD!  
Lower Case: hello, world!  
Length: 13  
Contains 'World': True  
Replaced Text: Hi, World!
```

String manipulation using StringBuilder in C#:

using System.Text;

class Program

{

static void Main(string[] args)

{

StringBuilder sb = new StringBuilder();

sb.Append("Hello");

sb.Append(", ");

sb.Append("World!");

sb.Insert(5, "beautiful ");

sb.Replace("World", "Universe");

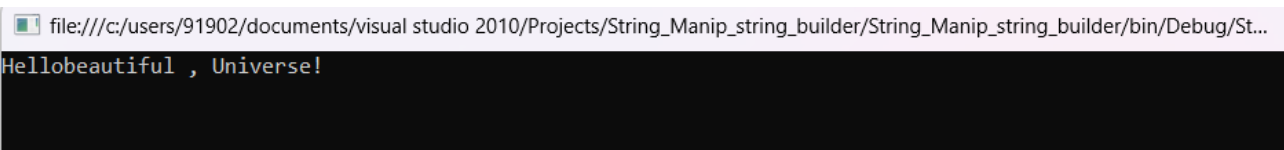
string finalText = sb.ToString();

Console.WriteLine(finalText);

Console.ReadLine();

}

}



file:///c:/users/91902/documents/visual studio 2010/Projects/String_Manip_string_builder/String_Manip_string_builder/bin/Debug/St...
Hellobeautiful , Universe!

Regex Matches method and regular expression pattern matching in C#:


```
using System;
using System.Text.RegularExpressions;

class Program
{
    static void Main(string[] args)
    {
        string input = "Hello, my email is example@example.com and my phone number is 123-456-7890.";

        // Match email addresses
        string emailPattern = @"\"b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\"b";
        MatchCollection emailMatches = Regex.Matches(input, emailPattern);
        foreach (Match match in emailMatches)
        {
            Console.WriteLine("Email: " + match.Value);
        }

        // Match phone numbers
        string phonePattern = @"\"b\d{3}-\d{3}-\d{4}\"b";
        MatchCollection phoneMatches = Regex.Matches(input, phonePattern);
        foreach (Match match in phoneMatches)
        {
            Console.WriteLine("Phone: " + match.Value);
        }

        Console.ReadLine();
    }
}
```



```
file:///c:/users/91902/documents/visual studio 2010/Projects/Regex_patternmathcing/Regex_patternmathcing/bin/Debug/Regex_pat...
Email: example@example.com
Phone: 123-456-7890
```


Unary Operator Overloading

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

namespace UnaryOperatorOverloading
{
    class Counter
    {
        public int Count { get; set; }

        public Counter(int count)
        {
            Count = count;
        }

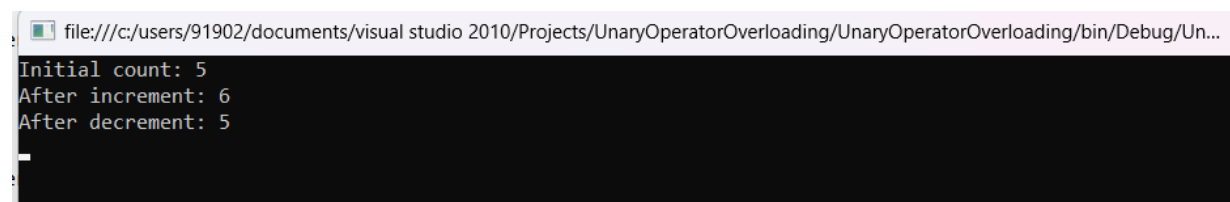
        public static Counter operator ++(Counter counter)
        {
            return new Counter(counter.Count + 1);
        }

        public static Counter operator --(Counter counter)
        {
            return new Counter(counter.Count - 1);
        }
    }

    class Program
    {
        static void Main()
        {
            Counter myCounter = new Counter(5);
            Console.WriteLine("Initial count: " + myCounter.Count);

            myCounter++; // Increment using unary operator
            Console.WriteLine("After increment: " + myCounter.Count);

            myCounter--; // Decrement using unary operator
            Console.WriteLine("After decrement: " + myCounter.Count);
            Console.ReadLine();
        }
    }
}
```



The screenshot shows a console window with the following output:

```
Initial count: 5
After increment: 6
After decrement: 5
```

Binary Operator Overloading

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

namespace BinaryOperatorOverloading
{
    class Vector
    {
        public int X { get; set; }
        public int Y { get; set; }

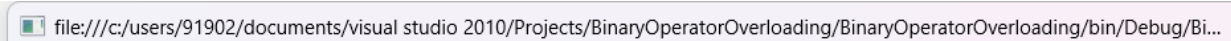
        public Vector(int x, int y)
        {
            X = x;
            Y = y;
        }

        public static Vector operator +(Vector v1, Vector v2)
        {
            int sumX = v1.X + v2.X;
            int sumY = v1.Y + v2.Y;
            return new Vector(sumX, sumY);
        }


        public override string ToString()
        {
            return "(" + X + ", " + Y + ")";
        }
    }
    class Program
    {
        static void Main()
        {
            Vector vector1 = new Vector(2, 3);
            Vector vector2 = new Vector(4, 5);

            Vector sum = vector1 + vector2; // Adding two Vector objects using binary operator

            Console.WriteLine("Vector 1: " + vector1);
            Console.WriteLine("Vector 2: " + vector2);
            Console.WriteLine("Sum: " + sum);
            Console.ReadLine();
        }
    }
}
```



file:///c:/users/91902/documents/visual studio 2010/Projects/BinaryOperatorOverloading/BinaryOperatorOverloading/bin/Debug/Bi...



```
Vector 1: (2, 3)
Vector 2: (4, 5)
Sum: (6, 8)
```

Window Based Student Registration Form With Validations

```
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.Text.RegularExpressions;

namespace WindowFormStudentRegistration
{
    public partial class Form1 : Form
    {
        string emailpattern = "[a-z0-9!#$%&'*/+=?^_`{|}~-]+(?:\\.[a-z0-9!#$%&'*/+=?^_`{|}~-]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?$";

        string passpattern = @"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[!@#%&*]).{8,}$";

        public Form1()
        {
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
        {
        }

        private void label3_Click(object sender, EventArgs e)
        {
        }

        private void button2_Click(object sender, EventArgs e)
        {
            txtID.Clear();
            txtName.Clear();
            txtEmail.Clear();
            txtPass.Clear();
            txtConPass.Clear();
            txtID.Focus();
        }

        private void txtID_Leave(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(txtID.Text) == true)
            {
                txtID.Focus();
                errorProvider1.SetError(this.txtID, "Please Fill Data");
            }
            else
        }
```

```

        {
            errorProvider1.Clear();
        }
    }

private void txtID_KeyPress(object sender, KeyPressEventArgs e)
{
    char ch = e.KeyChar;
    if (char.IsDigit(ch) == true)
    {
        e.Handled = false;
    }
    else if (ch == 8)
    {
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
    }
}

private void txtName_Leave(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtName.Text) == true)
    {
        txtName.Focus();
        errorProvider2.SetError(this.txtName, "Please Fill Data");
    }
    else
    {
        errorProvider2.Clear();
    }
}

private void txtName_KeyPress(object sender, KeyPressEventArgs e)
{
    char ch = e.KeyChar;
    if (char.IsLetter(ch) == true)
    {
        e.Handled = false;
    }
    else if (ch == 8)
    {
        e.Handled = false;
    }
    else if (ch == 32)
    {
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
    }
}

```

```

private void txtEmail_Leave(object sender, EventArgs e)
{
    if (Regex.IsMatch(txtEmail.Text, emailpattern) == false)
    {
        txtEmail.Focus();
        errorProvider3.SetError(this.txtEmail, "Please Enter Valid Data");
    }
    else
    {
        errorProvider3.Clear();
    }
}

private void txtPass_Leave(object sender, EventArgs e)
{
    if (Regex.IsMatch(txtPass.Text, passpattern) == false)
    {
        txtPass.Focus();
        errorProvider4.SetError(this.txtPass, "Please Enter Strong Password");
    }
    else
    {
        errorProvider4.Clear();
    }
}

private void txtConPass_Leave(object sender, EventArgs e)
{
    if (txtConPass.Text != txtPass.Text)
    {
        txtConPass.Focus();
        errorProvider5.SetError(this.txtConPass, "Enter Same Password");
    }
    else
    {
        errorProvider5.Clear();
    }
}

private void button1_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtID.Text) == true)
    {
        txtID.Focus();
        errorProvider1.SetError(this.txtID, "Please Fill Data");
    }
    else if (string.IsNullOrEmpty(txtName.Text) == true)
    {
        txtName.Focus();
        errorProvider2.SetError(this.txtName, "Please Fill Data");
    }
    else if (Regex.IsMatch(txtEmail.Text, emailpattern) == false)
    {
        txtEmail.Focus();
    }
}

```

```

        errorProvider3.SetError(this.txtEmail, "Please Enter Valid Data");
    }
    else if (Regex.IsMatch(txtPass.Text, passpattern) == false)
    {
        txtPass.Focus();
        errorProvider4.SetError(this.txtPass, "Please Enter Strong Password");
    }
    else if (txtConPass.Text != txtPass.Text)
    {
        txtConPass.Focus();
        errorProvider5.SetError(this.txtConPass, "Enter Same Password");
        MessageBox.Show("Please Enter a Same Password");
    }
    else
    {
        MessageBox.Show("Congratulations Your Data Is Submitted");
    }
}
}
}

```

JJMCOE STUDENT REGISTRATION

ID: 20036

Name: Harshad Bujare

Class: ☐ FE ☐ SE ☒ TE ☐ BE

Email: harshadbujare2019@gmail.com

Password:

Confirm Password:

Submit Reset

JJMCOE STUDENT REGISTRATION

ID: 20036

Name: Harshad Bujare

Class: ☐ FE ☐ SE ☒ TE ☐ BE

Email: harshadbujare2019@gmail.com

Password:

Confirm Password:

Submit

OK

Congratulations Your Data Is Submitted

Form1

JJMCOE STUDENT REGISTRATION

ID

Name

Class ☐ FE ☐ SE ☒ TE ☐ BE

Email

Password

Confirm Password

Please Fill Data

Form1

JJMCOE STUDENT REGISTRATION

ID

Name

Class ☐ FE ☐ SE ☒ TE ☐ BE

Email

Password

Confirm Password

Please Fill Data


Form1

JJMCOE STUDENT REGISTRATION

ID

Name

Class ☐ FE ☐ SE ☒ TE ☐ BE

Email 

Password

Confirm Password

Please Enter Valid Data

Form1

JJMCOE STUDENT REGISTRATION

ID

Name

Class ☐ FE ☐ SE ☒ TE ☐ BE

Email

Password

Confirm Password

Please Enter a Same Password

OK

Window Based JJMCOE Library System Form With Validations and Database With CRUD Operation

```
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.Configuration;
using System.Data.OleDb;

namespace CRUDLibrary
{
    public partial class Form1 : Form
    {
        string cs = ConfigurationManager.ConnectionStrings["DBCS"].ConnectionString;
        public Form1()
        {
            InitializeComponent();

            private void label2_Click(object sender, EventArgs e)
            {

            }

            private void groupBox1_Enter(object sender, EventArgs e)
            {

            }

            private void label5_Click(object sender, EventArgs e)
            {

            }

            private void label6_Click(object sender, EventArgs e)
            {

            }

            private void label1_Click(object sender, EventArgs e)
            {

            }

            private void label7_Click(object sender, EventArgs e)
            {

            }

            private void textBox5_TextChanged(object sender, EventArgs e)
```

```

{
    OleDbConnection con = new OleDbConnection(cs);
    con.Open();
    string query = "select * from Library where name like @name + '% '";
    OleDbDataAdapter da = new OleDbDataAdapter(query, con);
    da.SelectCommand.Parameters.AddWithValue("@name", txtSearch.Text.Trim());
    DataTable dt = new DataTable();
    da.Fill(dt);

    if (dt.Rows.Count > 0)
    {
        dataGridView1.DataSource = dt;
    }
    else
    {
        MessageBox.Show("No Record Found");
        dataGridView1.DataSource = null;
    }
}

private void Form1_Load(object sender, EventArgs e)
{
}

private void btninsert_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(textID.Text) == true)
    {
        textID.Focus();
        errorProvider1.SetError(this.textID, "Please Enter ID");
    }
    else if (string.IsNullOrEmpty(textName.Text) == true)
    {
        textName.Focus();
        errorProvider2.SetError(this.textName, "Please Enter Name");
    }
    else if (string.IsNullOrEmpty(BookNo.Text) == true)
    {
        BookNo.Focus();
        errorProvider3.SetError(this.BookNo, "Please Enter Book Number");
    }
    else if (string.IsNullOrEmpty(description.Text) == true)
    {
        description.Focus();
        errorProvider4.SetError(this.description, "Please Enter Book Description");
    }
    else if (string.IsNullOrEmpty(returninfo.Text) == true)
    {
        returninfo.Focus();
        errorProvider5.SetError(this.returninfo, "Please Enter Book return details");
    }
    else
    {
        errorProvider1.Clear();
    }
}

```

```

errorProvider2.Clear();
errorProvider3.Clear();
errorProvider4.Clear();
errorProvider5.Clear();

OleDbConnection con = new OleDbConnection(cs);
con.Open();

string query2 = "select * from Library where id=@id";
OleDbCommand cmd2 = new OleDbCommand(query2, con);
cmd2.Parameters.AddWithValue("@id", textID.Text);
OleDbDataReader dr = cmd2.ExecuteReader();
if (dr.HasRows == true)
{
    MessageBox.Show(textID.Text + " ID has already taken...!!");
}
else
{
    string query = "insert into Library values(@id, @name, @number, @description,
@returninfo)";
    OleDbCommand cmd = new OleDbCommand(query, con);
    cmd.Parameters.AddWithValue("@id", textID.Text);
    cmd.Parameters.AddWithValue("@name", textName.Text);
    cmd.Parameters.AddWithValue("@number", BookNo.Value);
    cmd.Parameters.AddWithValue("@description", description.Text);
    cmd.Parameters.AddWithValue("@returninfo", returninfo.Text);

    int a = cmd.ExecuteNonQuery();
    if (a > 0)
    {
        MessageBox.Show("Record Saved Succesfully", "SUCCESS",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        display();
        reset();
    }
    else
    {
        MessageBox.Show("Failed", "Failed", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
    con.Close();
}
}
}

void reset()
{
    textID.Clear();
    textName.Clear();
    BookNo.Value = 0;
    description.Clear();
    returninfo.Clear();
    textID.Focus();
}

void display()

```

```

{
    OleDbConnection con = new OleDbConnection(cs);
    con.Open();
    string query = "select * from Library";
    OleDbDataAdapter da = new OleDbDataAdapter(query, con);
    DataTable dt = new DataTable();
    da.Fill(dt);
    dataGridView1.DataSource = dt;
}

private void btnreset_Click(object sender, EventArgs e)
{
    reset();
}

private void btmview_Click(object sender, EventArgs e)
{
    display();
}

private void dataGridView1_DoubleClick(object sender, EventArgs e)
{
    textID.Text = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();
    textName.Text = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();
    BookNo.Text = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();
    description.Text = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();
    returninfo.Text = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();
}

private void btnupdate_Click(object sender, EventArgs e)
{
    OleDbConnection con = new OleDbConnection(cs);
    con.Open();

    string query = "UPDATE Library set id=@id, name=@name, number=@number,
description=@description, returninfo=@returninfo WHERE id=@id";
    OleDbCommand cmd = new OleDbCommand(query, con);
    cmd.Parameters.AddWithValue("@id", textID.Text);
    cmd.Parameters.AddWithValue("@name", textName.Text);
    cmd.Parameters.AddWithValue("@number", BookNo.Value);
    cmd.Parameters.AddWithValue("@description", description.Text);
    cmd.Parameters.AddWithValue("@returninfo", returninfo.Text);

    int a = cmd.ExecuteNonQuery();
    if (a > 0)
    {
        MessageBox.Show("Record updated Succesfully", "SUCCESS", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        display();
        reset();
    }
    else
    {
        MessageBox.Show("Failed", "Failed", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
}

```

```

    }
    con.Close();
}

private void btndelete_Click(object sender, EventArgs e)
{
    OleDbConnection con = new OleDbConnection(cs);
    con.Open();

    string query = "delete from Library where id=@id";
    OleDbCommand cmd = new OleDbCommand(query, con);
    cmd.Parameters.AddWithValue("@id", textID.Text);

    int a = cmd.ExecuteNonQuery();
    if (a > 0)
    {
        MessageBox.Show("Record Deleted Succesfully", "SUCCESS", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        display();
        reset();
    }
    else
    {
        MessageBox.Show("Failed", "Failed", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
    con.Close();
}

private void search_Click(object sender, EventArgs e)
{
    OleDbConnection con = new OleDbConnection(cs);
    con.Open();
    string query = "select * from Library where name like @name + '%'";
    OleDbDataAdapter da = new OleDbDataAdapter(query, con);
    da.SelectCommand.Parameters.AddWithValue("@name", txtSearch.Text.Trim());
    DataTable dt = new DataTable();
    da.Fill(dt);

    if (dt.Rows.Count > 0)
    {
        dataGridView1.DataSource = dt;
    }
    else
    {
        MessageBox.Show("No Record Found");
        dataGridView1.DataSource = null;
    }
}
}
}

```

Library System

JJMCOE STUDENT LIBRARY SYSTEM

Book ID: 30037
Book Name: Computer Algorithm
Book No.: 12
Description: CA Book of Sartai Sahani
Return Info: After 10 D

Insert
Update
Delete
View
Reset

Search:

Record Saved Successfully

OK

Library System

JJMCOE STUDENT LIBRARY SYSTEM

Book ID:
Book Name:
Book No.: 0
Description:
Return Info:

Insert
Update
Delete
View
Reset

Search:

	id	name	number	description	returninfo
▶	30037	Computer Algorithm	12	CA Book of Sarta...	After 10 Days

Library System

JJMCOE STUDENT LIBRARY SYSTEM

Book ID:
Book Name:
Book No.: 0
Description:
Return Info:

Insert
Update
Delete
View
Reset

Search: Co

	id	name	number	description	returninfo
▶	30037	Computer Algorithm	12	CA Book of Sarta...	After 10 Days

```

CRUDLibrary (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team XML Data Tools Test Window Help
App.config Form1.cs Form1.cs [Design] Program.cs Form1.resx Form1.Designer.cs
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <connectionStrings>
    <add name="DBCS" connectionString="Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Users\91902\Documents\Database1.accdb" providerName="System.Data.OleDb"/>
  </connectionStrings>
</configuration>

```

JJMCOE STUDENT LIBRARY SYSTEM

Book ID

Book Name

Book No.

Description

Return Info

Insert

Please Enter ID
Update

Delete

View

Reset

Search

id	name	number	description	returninfo
30037	Computer Algorithm	12	CA Book of Sarta...	After 10 Days
30038	C# Programing	15	C# programing by...	After 15 Days

JJMCOE STUDENT LIBRARY SYSTEM

Book ID

Book Name

Book No.

Description

Return Info

Insert

Update

Delete

View

Please Enter Book Description
Reset

Search

Library System

Book ID

30040

Insert

Book Name

Database

Update

Book No.

20

Delete

Description

DB by john mathew

View

Return Info

after 12 days Monday

Reset

Search

id	name	number	description	returninfo
30037	Computer Algorithm	12	CA Book of Sarta...	After 10 Days
30038	C# Programing	15	C# programing by...	After 15 Days
30039	Operating System	18	OS by Amruta rat...	After 2 days
30040	Database	20	DB by john mathew	after 12 days Mo...

Library System

Book ID

30040

Insert

Book Name

Database

Update

Book No.

20

Delete

Description

DB by john mathew

View

Return Info

after 12 d

Reset

Search

id	name	number	description	returninfo
30037	Computer Algorithm	12	CA Book of Sarta...	After 10 Days
30038	C# Programing	15	C# programing by...	After 15 Days
30039	Operating System	18	OS by Amruta rat...	After 2 days
30040	Database	20	DB by john mathew	after 12 days Mo...

Library System

JJMCOE STUDENT LIBRARY SYSTEM

Book ID
Book Name
Book No. 0
Description
Return Info

Insert
Update
Delete
View
Reset

Search

id	name	number	description	returninfo
30037	Computer Algorithm	12	CA Book of Sarta...	After 10 Days
30038	C# Programing	15	C# programing by...	After 15 Days
30039	Operating System	18	OS by Amruta rat...	After 2 days

Database1 : Database- C:\Users\91902\Documents\Database1.accdb (Access 2007 - 2016 file format) - Access

id	name	number	description	returninfo
30037	Computer Algor 12	12	CA Book of Sarti	After 10 Days
30038	C# Programing	15	C# programing I	After 15 Days
30039	Operating Syste 18	18	OS by Amruta r	After 2 days

CRUDLibrary (Running) - Microsoft Visual Studio

```

else
{
    MessageBox.Show("No Record Found");
    dataGridView1.DataSource = null;
}

private void Form1_Load(object sender, EventArgs e)
{
}

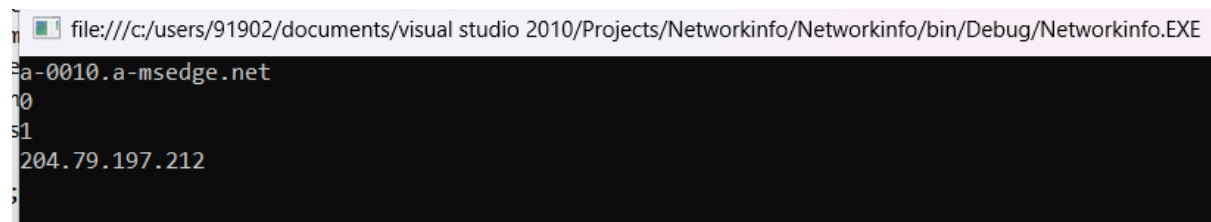
private void btninsert_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(textID.Text) == true)
    {
        textID.Focus();
        errorProvider1.SetError(this.textID, "Please Enter ID");
    }
    else if (string.IsNullOrEmpty(textName.Text) == true)
    {
        textName.Focus();
        errorProvider2.SetError(this.textName, "Please Enter Name");
    }
    else if (string.IsNullOrEmpty(BookNo.Text) == true)
    {
        BookNo.Focus();
        errorProvider3.SetError(this.BookNo, "Please Enter Book Number");
    }
    else if (string.IsNullOrEmpty(description.Text) == true)
    {
        description.Focus();
        errorProvider4.SetError(this.description, "Please Enter Book Description");
    }
    else if (string.IsNullOrEmpty(returninfo.Text) == true)
    {
        returninfo.Focus();
        errorProvider5.SetError(this.returninfo, "Please Enter Book return details");
    }
}

```

Console Based Network application

To Obtain Information about Network Part 1

```
using System;
using System.Net;
using System.Net.Sockets;
class MyClient
{
    public static void Main()
    {
        IPEndPoint IPEndPoint = Dns.GetHostEntry("www.hotmail.com");
        Console.WriteLine(IPEndPoint.HostName);
        string[] aliases = IPEndPoint.Aliases;
        Console.WriteLine(aliases.Length);
        IPAddress[] addr = IPEndPoint.AddressList;
        Console.WriteLine(addr.Length);
        for (int i = 0; i < addr.Length; i++)
        {
            Console.WriteLine(addr[i]);
        }
        Console.ReadLine();
    }
}
```



To Obtain Information about Network Part 2

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

class NetDetails
{
    static void Main(string[] args)
    {
        // Get an array of all network interfaces on the machine
        NetworkInterface[] niArr = NetworkInterface.GetAllNetworkInterfaces();

        Console.WriteLine("Retrieving basic information of network.\n\n");

        // Iterate over each network interface
```

```

foreach (NetworkInterface tempNetworkInterface in niArr)
{
    // Print the network interface information
    Console.WriteLine("Network Description: " + tempNetworkInterface.Description);
    Console.WriteLine("Network ID: " + tempNetworkInterface.Id);
    Console.WriteLine("Network Name: " + tempNetworkInterface.Name);
    Console.WriteLine("Network Interface Type: " +
tempNetworkInterface.NetworkInterfaceType.ToString());
    Console.WriteLine("Network Operational Status: " +
tempNetworkInterface.OperationalStatus.ToString());
    Console.WriteLine("Network Speed: " + tempNetworkInterface.Speed);
    Console.WriteLine("Supports Multicast: " + tempNetworkInterface.SupportsMulticast);
    Console.WriteLine();
}
Console.ReadLine();
}
}

```

```

Retrieving basic information of network.

Network Description: Realtek PCIe GbE Family Controller
Network ID: {B3009843-199D-4A27-A73E-E66E8138EB86}
Network Name: Ethernet
Network Interface Type: Ethernet
Network Operational Status: Down
Network Speed: -1
Supports Multicast: True

Network Description: TAP-Windows Adapter V9
Network ID: {E87E00A6-4393-41E0-809F-8B47FDC4F9CE}
Network Name: Local Area Connection
Network Interface Type: 53
Network Operational Status: Down
Network Speed: 1000000000
Supports Multicast: True

Network Description: Microsoft Wi-Fi Direct Virtual Adapter #3
Network ID: {9AFD6978-7516-4C6C-9203-0E6C8AA75761}
Network Name: Local Area Connection* 2
Network Interface Type: Wireless80211
Network Operational Status: Down
Network Speed: -1
Supports Multicast: True

Network Description: Microsoft Wi-Fi Direct Virtual Adapter #4
Network ID: {3219E35E-2B38-4A3B-AD20-79AE071FD88C}
Network Name: Local Area Connection* 3
Network Interface Type: Wireless80211
Network Operational Status: Down
Network Speed: -1
Supports Multicast: True

Network Description: VMware Virtual Ethernet Adapter for VMnet1
Network ID: {F7647AC1-6869-4AF5-8742-178298C7EFEB}
Network Name: VMware Network Adapter VMnet1
Network Interface Type: Ethernet
Network Operational Status: Up
Network Speed: 1000000000
Supports Multicast: True

Network Description: VMware Virtual Ethernet Adapter for VMnet8
Network ID: {2592F4C9-CFEA-41C1-AE85-92FE76C67C99}
Network Name: VMware Network Adapter VMnet8
Network Interface Type: Ethernet
Network Operational Status: Up
Network Speed: 1000000000
Supports Multicast: True

Network Description: Intel(R) Wi-Fi 6 AX201 160MHz

```

To Detect Changes in Network

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net.NetworkInformation;

```

```

class netchange
{
    static void Main(string[] args)
    {
        // Add the handlers to the NetworkChange events.
    }
}

```

```

NetworkChange.NetworkAvailabilityChanged += NetworkAvailabilityChanged;
NetworkChange.NetworkAddressChanged += NetworkAddressChanged;

Console.ReadLine();
}

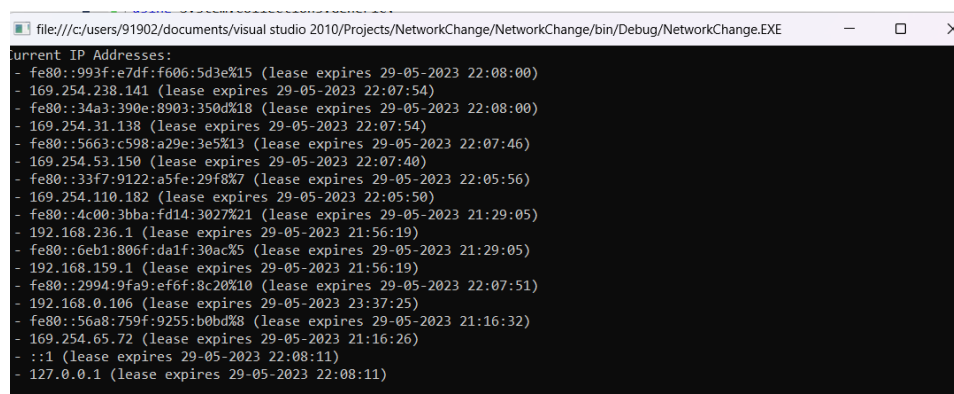
// Declare a method to handle NetworkAvailabilityChanged events.
private static void NetworkAvailabilityChanged(object sender, NetworkAvailabilityEventArgs e)
{
    // Report whether the network is now available or unavailable.
    if (e.IsAvailable)
    {
        Console.WriteLine("Network Available");
    }
    else
    {
        Console.WriteLine("Network Unavailable");
    }
}

// Declare a method to handle NetworkAddressChanged events.
private static void NetworkAddressChanged(object sender, EventArgs e)
{
    Console.WriteLine("Current IP Addresses:");

    // Iterate through the interfaces and display information.
    foreach (NetworkInterface ni in NetworkInterface.GetAllNetworkInterfaces())
    {
        foreach (UnicastIPAddressInformation addr in ni.GetIPProperties().UnicastAddresses)
        {
            Console.WriteLine("- {0} (lease expires {1})", addr.Address, DateTime.Now + new
            TimeSpan(0, 0, (int)addr.DhcpLeaseLifetime));
        }
    }
    Console.ReadLine();
}
}

```

After Close the WIFI Connection



```

file:///c:/users/91902/documents/visual studio 2010/Projects/NetworkChange/NetworkChange/bin/Debug/NetworkChange.EXE
current IP Addresses:
- fe80::993f:e7df:f606:5d3e%15 (lease expires 29-05-2023 22:08:00)
- 169.254.238.141 (lease expires 29-05-2023 22:07:54)
- fe80::34a3:390e:8903:350d%18 (lease expires 29-05-2023 22:08:00)
- 169.254.31.138 (lease expires 29-05-2023 22:07:54)
- fe80::5663:c598:a29e:3e5%13 (lease expires 29-05-2023 22:07:46)
- 169.254.53.150 (lease expires 29-05-2023 22:07:40)
- fe80::33f7:9122:a5fe:29f8%7 (lease expires 29-05-2023 22:05:56)
- 169.254.110.182 (lease expires 29-05-2023 22:05:50)
- fe80::4c00:3bba:fd14:3027%21 (lease expires 29-05-2023 21:29:05)
- 192.168.236.1 (lease expires 29-05-2023 21:56:19)
- fe80::6eb1:806f:da1f:30ac%5 (lease expires 29-05-2023 21:29:05)
- 192.168.159.1 (lease expires 29-05-2023 21:56:19)
- fe80::2994:9fa9:ef6f:8c20%10 (lease expires 29-05-2023 22:07:51)
- 192.168.0.106 (lease expires 29-05-2023 23:37:25)
- fe80::56a8:759f:9255:b0bd%3 (lease expires 29-05-2023 21:16:32)
- 169.254.65.72 (lease expires 29-05-2023 21:16:26)
- ::1 (lease expires 29-05-2023 22:08:11)
- 127.0.0.1 (lease expires 29-05-2023 22:08:11)

```