EXP-1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication5

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Hi!");

Console.ReadLine();

}

}

}

EXP-2

//Class Library

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassLibrary2

{

public class Class2

{

public string Display()

{

return ("I m in Display");

}

public string Print()

{

return ("I m in Print");

}

}

}

Console Application

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using ClassLibrary2;

namespace ConsoleApp7

{

public static class XX

{

public static void NewMethod(this Class2 ob)

{

Console.WriteLine("Hello I m extended method");

}

}

class Program

{

static void Main(string[] args)

{

Class2 ob = new

Class2();

ob.Display();

ob.Print();

ob.NewMethod();

Console.ReadKey(

);

}

}

}

EXP-3

//Class Library

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassLibrary1

{

public class Class1

{

public int Addition(int x, int y)

{

return x + y;

}

public int Substraction(int x, int y)

{

return x - y;

}

public int Multiply(int x, int y)

{

return x \* y;

}

public int Divide(int x, int y)

{

return x / y;

}

}

}

Console Application

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using ClassLibrary1;

namespace exDLL

{

class Program

{

static void Main(string[] args)

{

Class1 ob = new Class1();

int sol = ob.Addition(10,79);

Console.WriteLine("The addition is:" + sol);

int ans = ob.Substraction(25, 90);

Console.WriteLine("The Substraction is:" + ans);

int res = ob.Multiply(22, 22);

Console.WriteLine("The Multiplication is:" + res);

int sol2 = ob.Divide(50, 20);

Console.WriteLine("The Division is:" + sol2); Console.ReadLine();

}

}

}

EXP-4

1

using System;

class Program

{

static void Main()

{

Console.WriteLine("Enter number of rows for the 2D array:");

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

Console.WriteLine("Enter number of columns for the 2D array:");

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

int[,] multiArray = new int[rows, cols];

Console.WriteLine("Enter elements for the 2D array:");

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

Console.Write("Enter element for position ({i},{j}):");

multiArray[i, j] = int.Parse(Console.ReadLine());

}

}

Console.WriteLine("\nMultidimensional Array (2D):");

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

Console.Write(multiArray[i, j] + " ");

}

Console.WriteLine();

}

Console.ReadLine();

}

}

2

using System;

class Program

{

static void Main()

{

Console.WriteLine("Enter the number of rows for the jagged array:");

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

int[][] jaggedArray = new int[rows][];

for (int i = 0; i < rows; i++)

{

Console.WriteLine("Enter the number of elements for row {[i] + 1}:");

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

jaggedArray[i] = new int[cols];

for (int j = 0; j < cols; j++)

{

Console.Write("Enter element for position ({i},{j}):");

jaggedArray[i][j] = int.Parse(Console.ReadLine());

}

}

Console.WriteLine("\nJagged Array:");

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < jaggedArray[i].Length; j++)

{

Console.Write(jaggedArray[i][j] + " ");

}

Console.WriteLine();

}

Console.ReadLine();

}

}

EXP-5

1

using System;

class Student

{

public string Name

{

get;

set;

}

public int MathMarks

{

get;

set;

}

public int ScienceMarks

{

get;

set;

}

public int EnglishMarks

{

get;

set;

}

public Student(string name, int mathMarks, int scienceMarks, int englishMarks)

{

Name = name;

MathMarks = mathMarks;

ScienceMarks = scienceMarks;

EnglishMarks = englishMarks;

}

public static Student operator +(Student student1, Student student2)

{

int combinedMathMarks = student1.MathMarks + student2.MathMarks;

int combinedScienceMarks = student1.ScienceMarks + student2.ScienceMarks;

int combinedEnglishMarks = student1.EnglishMarks + student2.EnglishMarks;

return new Student( student1.Name + " & " + student2.Name, combinedMathMarks, combinedScienceMarks, combinedEnglishMarks );

}

public void Display()

{

Console.WriteLine("Name:"+ Name);

Console.WriteLine("Math Marks:"+ MathMarks);

Console.WriteLine("Science Marks:"+ ScienceMarks);

Console.WriteLine("English Marks:"+ EnglishMarks);

Console.WriteLine("Total Marks: "+ MathMarks + ScienceMarks + EnglishMarks);

Console.WriteLine(" ");

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter details for Student 1:");

Console.Write("Name: "); string name1 = Console.ReadLine();

Console.Write("Math Marks: ");

int mathMarks1 = Convert.ToInt32(Console.ReadLine());

Console.Write("Science Marks: ");

int scienceMarks1 = Convert.ToInt32(Console.ReadLine());

Console.Write("English Marks: ");

int englishMarks1 = Convert.ToInt32(Console.ReadLine());

Student student1 = new Student(name1, mathMarks1, scienceMarks1, englishMarks1);

Console.WriteLine("\nEnter details for Student 2:");

Console.Write("Name: ");

string name2 = Console.ReadLine();

Console.Write("Math Marks: ");

int mathMarks2 = Convert.ToInt32(Console.ReadLine());

Console.Write("Science Marks: ");

int scienceMarks2 = Convert.ToInt32(Console.ReadLine());

Console.Write("English Marks: ");

int englishMarks2 = Convert.ToInt32(Console.ReadLine());

Student student2 = new Student(name2, mathMarks2, scienceMarks2, englishMarks2);

Console.WriteLine("\nStudent 1 details:");

student1.Display();

Console.WriteLine("Student 2 details:");

student2.Display();

Student combinedStudent = student1 + student2;

Console.WriteLine("Combined Student details:");

combinedStudent.Display();

Console.ReadLine();

}

}

2

using System;

class Program

{

static void Main()

{

string str1 = "Hello";

string str2 = "World";

string result = str1 + " " + str2;

Console.WriteLine(result);

string interpolated = str1 + ""+str2;

Console.WriteLine(interpolated );

string sub = result.Substring(0, 5);

Console.WriteLine(sub);

string replaced = result.Replace("World", "C#");

Console.WriteLine(replaced);

string[] words = result.Split(' ');

Console.WriteLine(words[0]);

string joined = string.Join("-", words);

Console.WriteLine(joined);

Console.WriteLine(result.ToUpper());

Console.WriteLine(result.ToLower());

string spaced = " Trim me ";

Console.WriteLine(spaced.Trim());

Console.ReadLine();

}

}

3

using System;

using System.Text;

class Program

{

static void Main()

{

StringBuilder sb = new StringBuilder("Hello");

sb.Append(" World");

Console.WriteLine(sb);

sb.Insert(5, ",");

Console.WriteLine(sb);

sb.Replace("World", "C#");

Console.WriteLine(sb);

sb.Remove(5, 1);

Console.WriteLine(sb);

sb.Clear();

sb.Append("New String");

Console.WriteLine(sb);}

}

EXP-6

Windows Form

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_6

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void newToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form2 newMDIChild = new Form2();

newMDIChild.MdiParent = this;

newMDIChild.Show();

}

}

}

EXP-7

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace WindowsFormsApplication3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection();

con.ConnectionString = @"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\USERS\NIHAR SATISH BAMBARE\ONEDRIVE\DESKTOP\C#\WINDOWSFORMSAPPLICATION3\WINDOWSFORMSAPPLICATION3\DATABASE1.MDF;Integrated Security=True";

try

{

con.Open();

SqlCommand insertCommand = new SqlCommand("INSERT INTO [Table] (Vname, Vcolor) VALUES (@Vname, @Vcolor)", con);

insertCommand.Parameters.AddWithValue("@Vname", textBox1.Text);

insertCommand.Parameters.AddWithValue("@Vcolor", textBox2.Text);

insertCommand.ExecuteNonQuery();

SqlCommand selectCommand = new SqlCommand("SELECT \* FROM [Table]", con);

SqlDataAdapter dAdapter = new SqlDataAdapter(selectCommand);

DataSet ds = new DataSet();

dAdapter.Fill(ds);

dataGridView1.ReadOnly = true;

dataGridView1.DataSource = ds.Tables[0];

MessageBox.Show("Record Submitted", "Congrats");

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message, "Error");

}

finally

{

con.Close();

}

}

}

}EXP-8

EXP-9

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Net.NetworkInformation;

namespace Networking\_Program

{

class Program

{

static void Main(string[] args)

{

var nics = from nic in NetworkInterface.GetAllNetworkInterfaces()where nic.OperationalStatus== OperationalStatus.Up select nic;

foreach (var nic in nics)

{

Console.WriteLine("Network ID: " + nic.Id);

Console.WriteLine("Network Name: " + nic.Name);

Console.WriteLine("Network Desription: " + nic.Description);

Console.WriteLine("Network Interface Type: " + nic.NetworkInterfaceType);

Console.WriteLine("Network Operational Status: " + nic.OperationalStatus);

Console.WriteLine("Network Speed: " + nic.Speed);

Console.WriteLine();

}

Console.ReadLine();

}

}

}

EXP-10

m id="form1" runat="server">

<asp:Label ID="Label1" runat="server" Text="Enter Your Name"></asp:Label>

<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"

ControlToValidate="TextBox1" ErrorMessage="Enter Something"></asp:RequiredFieldValidator>

<p>

<asp:Label ID="Label2" runat="server" Text="Enter Your Surname"></asp:Label>

<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"

ControlToValidate="TextBox2" ErrorMessage="Enter Something"></asp:RequiredFieldValidator>

</p>

<p>

<asp:Label ID="Label7" runat="server" Text="Enter Your Roll No"></asp:Label>

<asp:TextBox ID="TextBox7" runat="server"></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator7" runat="server"

ControlToValidate="TextBox7" ErrorMessage="Enter Something"></asp:RequiredFieldValidator>

</p>

<p>

<asp:Label ID="Label4" runat="server" Text="Enter Mobile No"></asp:Label>

<asp:TextBox ID="TextBox4" runat="server"></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server"

ControlToValidate="TextBox4" ErrorMessage="Enter Something"></asp:RequiredFieldValidator>

<asp:RangeValidator ID="RangeValidator1" runat="server" ErrorMessage="Should be

10 digit" MaximumValue="10" MinimumValue="10"></asp:RangeValidator>

</p>

<p>

<asp:Label ID="Label8" runat="server" Text="Enter Email"></asp:Label>

<asp:TextBox ID="TextBox5" runat="server" Height="16px"

Width="126px"></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator8" runat="server"

ControlToValidate="TextBox5" ErrorMessage="Enter Something"></asp:RequiredFieldValidator>

<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server"

ControlToValidate="TextBox5" ErrorMessage="Enter Correct Email"

ValidationExpression="\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-

.]\w+)\*"></asp:RegularExpressionValidator>

</p>

<p>

<asp:Label ID="Label6" runat="server" Text="Enter Address"></asp:Label>

<asp:TextBox ID="TextBox6" runat="server"></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator6" runat="server"

ControlToValidate="TextBox6" ErrorMessage="Enter Something"></asp:RequiredFieldValidator>

</p>

<p>

<asp:Button ID="Button1" runat="server" Text="Submit" />

</p>

</form>