1. **Describe the enumerations programming constructs, which provides a human-readable form of a series of related constant values in C#..**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace EnumerationDemo

{

class ProgramOne

{

enum Colors

{

Green,

Blue,

Yellow,

Violet,

Red,

Orange,

Pink

}

static void Main(string[] args)

{

foreach (var color in Enum.GetValues(typeof(Colors)))

{

Console.WriteLine("{0} : {1}", color, (int)color);

}

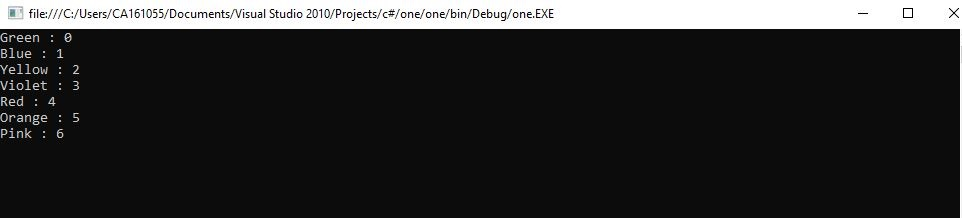
Console.Read();

}

}

}

**OUTPUT**



1. **Check Whether the Entered Year is a Leap Year or Not**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ProgramTwo

{

class Program

{

static void Main(string[] args)

{

try {

Console.Write("Enter The Year : \n");

long year = Convert.ToInt64(Console.ReadLine());

Console.WriteLine("\n-----------------------------------------");

if (year % 400 == 0) {

Console.WriteLine("\t{0} is a Leap Year",year);

}

else if (year % 100 == 0) {

Console.WriteLine("\t{0} is not a Leap Year", year);

}

else if (year % 4 == 0)

{

Console.WriteLine("\t{0} is a Leap Year", year);

}

else {

Console.WriteLine("\t{0} is not a Leap Year", year);

}

}

catch(Exception ex) {

Console.WriteLine("Enter valid year");

}

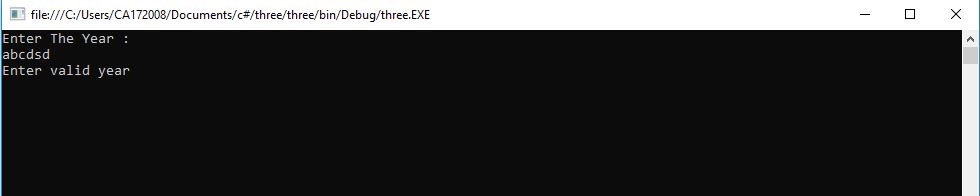
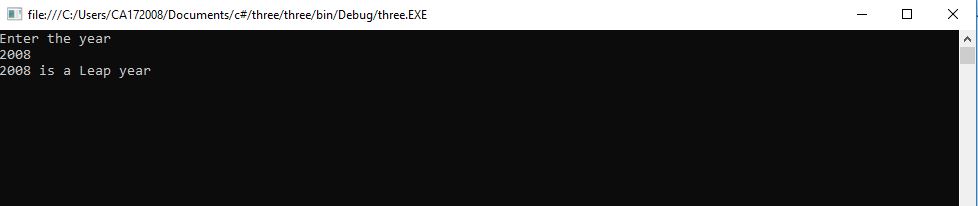
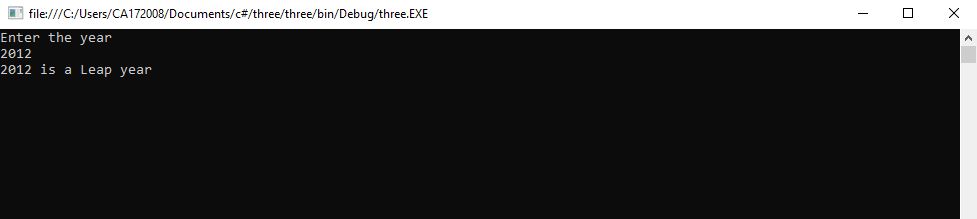
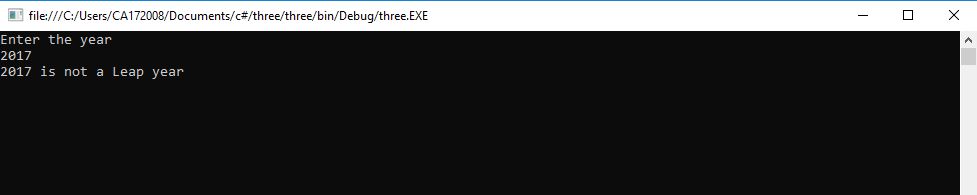
Console.ReadKey();

}

}

}

**OUTPUT**



1. **Program to display the addition using the windows application.**

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 three

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

int num1 = Int16.Parse(textBox1.Text);

int num2 = Int16.Parse(textBox2.Text);

int sum = num1 + num2;

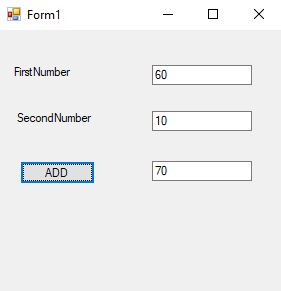
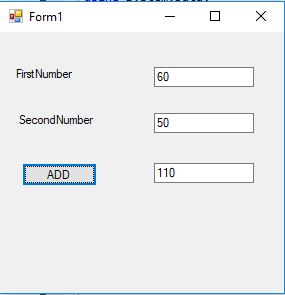
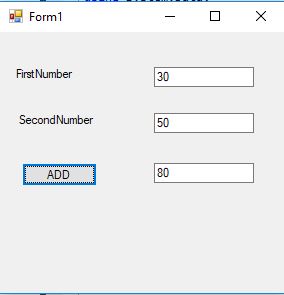
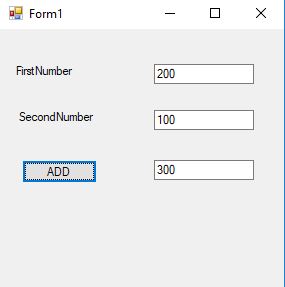
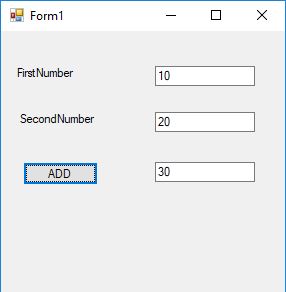
textBox3.Text = "sum of two number :" + sum;

}

}

}

**OUTPUT**



1. **Program to display the addition, subtraction, multiplication and division of two number using console applications.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ProgramFive

{

class Program

{

static void Main(string[] args)

{

double num1, num2;

double sum, sub, mul, div;

Console.WriteLine("Enter the two numbers");

num1 = Double.Parse(Console.ReadLine());

num2 = Double.Parse(Console.ReadLine());

sum = num1 + num2;

sub = num1 - num2;

mul = num1 \* num2;

div = num1 / num2;

Console.WriteLine();

Console.WriteLine("----------------------------");

Console.WriteLine("Addition: {0}", sum);

Console.WriteLine("Substraction: {0}", sub);

Console.WriteLine("Multiplication: {0}", mul);

Console.WriteLine("Division: {0}", div);

Console.WriteLine("----------------------------");

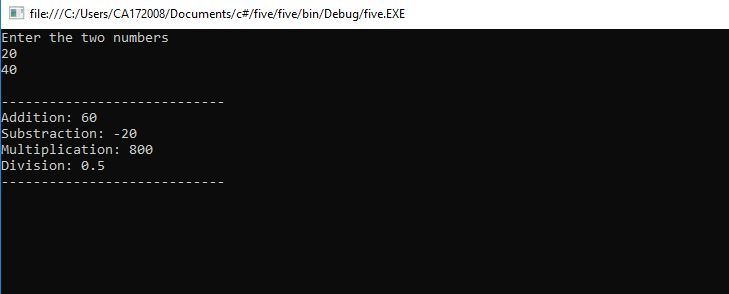
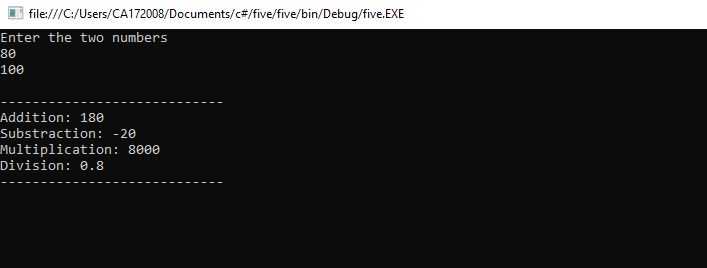
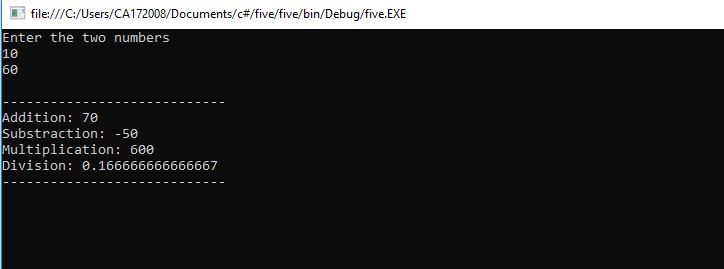
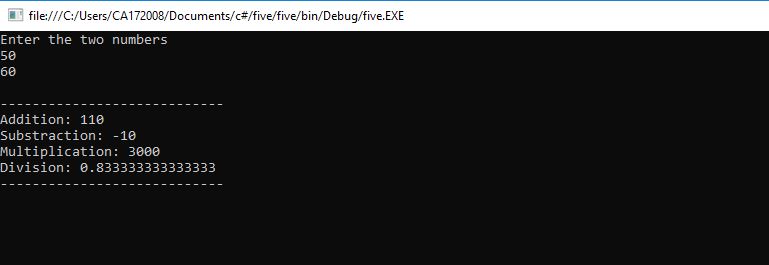
Console.ReadLine();

}

}

}

**OUTPUT**



1. **Program to display the first 10 natural numbers and their sum using console application**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ProgramSix

{

class Program

{

static void Main(string[] args)

{

int sum = 0;

Console.WriteLine("------------------");

Console.WriteLine("First 10 natural numbers");

Console.WriteLine("------------------");

for (int i = 1; i <= 10; i++)

{

sum += i;

Console.WriteLine(i);

}

Console.WriteLine("------------------");

Console.WriteLine("Sum: {0}", sum);

Console.WriteLine("------------------");

Console.ReadLine();

}

}

}

**OUTPUT**



1. **Write a program to convert input string from lower to upper and upper to lower case.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace LowUp

{

class Program

{

static void Main(string[] args)

{

string str1;

char[] arr1;

int l,i;

l=0;

char ch;

Console.Write("Input the string : ");

str1 = Console.ReadLine();

l=str1.Length;

arr1 = str1.ToCharArray(0, l);

Console.Write("\nAfter conversion, the string is : ");

for (i = 0; i < l; i++)

{

ch = arr1[i];

if (Char.IsLower(ch))

Console.Write(Char.ToUpper(ch));

else

Console.Write(Char.ToLower(ch));

}

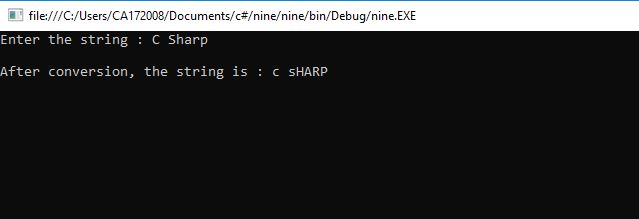
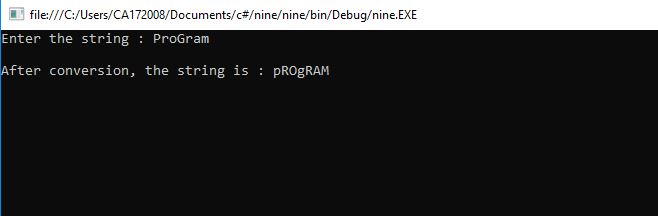
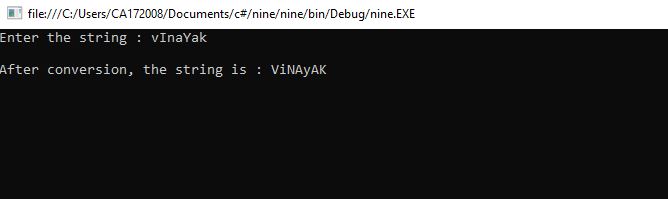
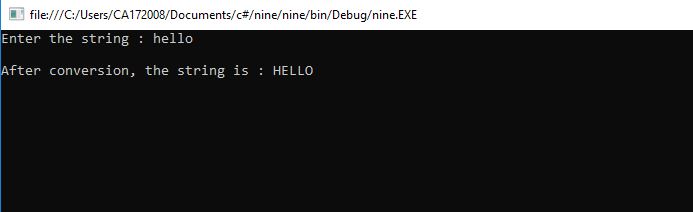
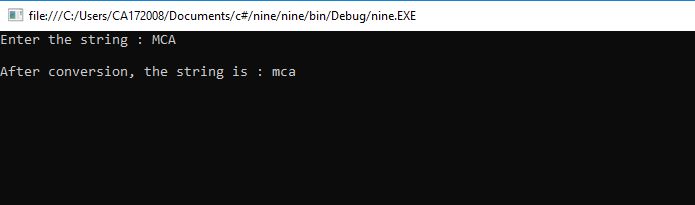
Console.ReadLine();

}

}

}

**OUTPUT**



1. **Demonstrate Command line arguments processing.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Sixteen

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("\nNumber of CommadLine Arguments :" + args.Length);

Console.Write("\nCommandline Arguments Are :\t");

for (int i = 0; i < args.Length; i++)

{

Console.Write(args[i] + "\t");

}

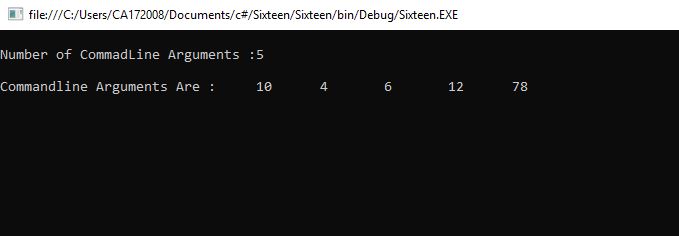
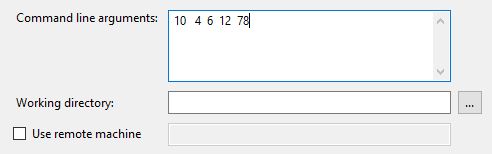
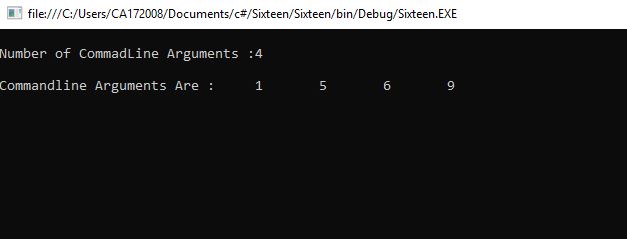
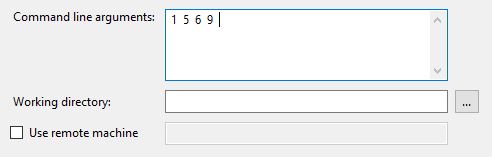
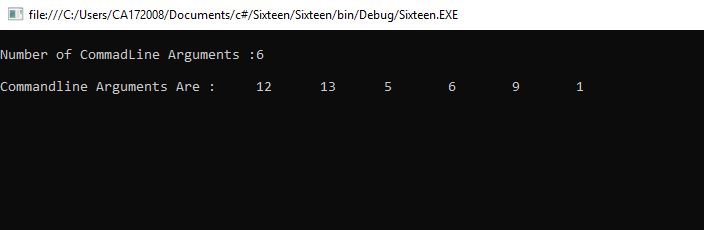
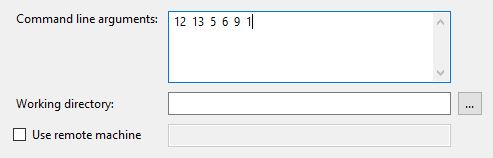
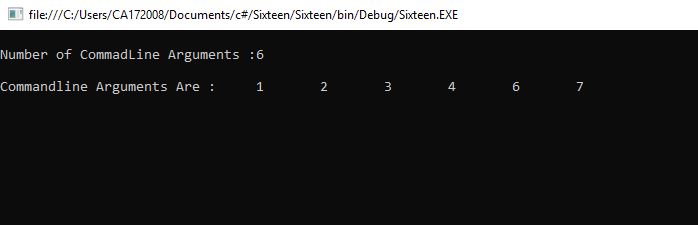
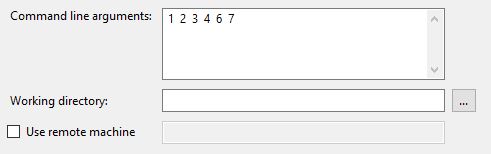
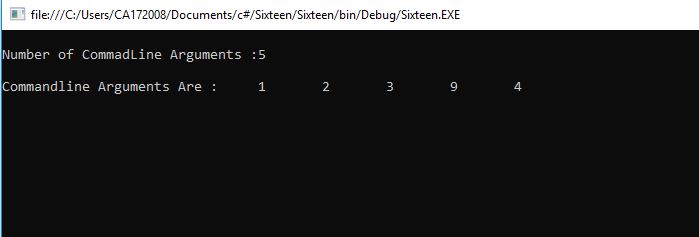
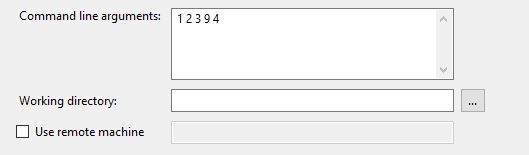
Console.ReadLine();

}

}

}

**OUTPUT**



1. **Find the second largest element in a single dimensional array.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace pgm14

{

class Program

{

static void Main(string[] args)

{

try

{

int[] arr = new int[5];

Console.WriteLine("Enter 5 element in array : ");

for (int i = 0; i < 5; i++)

{

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

}

Console.WriteLine("---------------------");

Array.Sort(arr);

Array.Reverse(arr);

Console.WriteLine("Sorted Array in Reverce Order");

for (int i = 0; i < 5; i++)

{

Console.WriteLine("A[" + i + "] = " + arr[i]);

}

Console.WriteLine("Second Largest Value in Array : " + arr[1]);

}

catch (Exception ex) {

Console.WriteLine("Provide Valid Array Element.\nOnly Numeric Values are

allowed.");

}

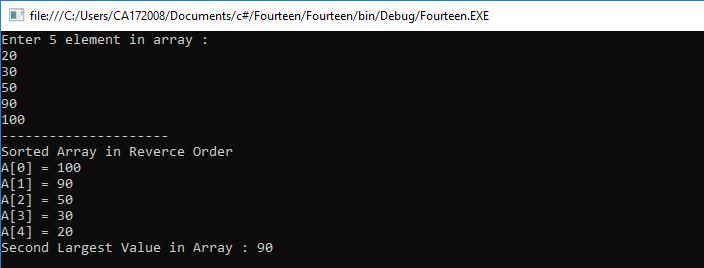
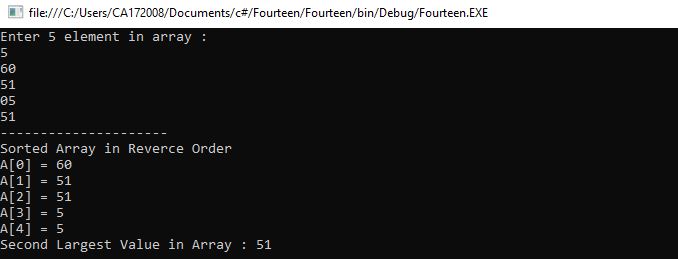
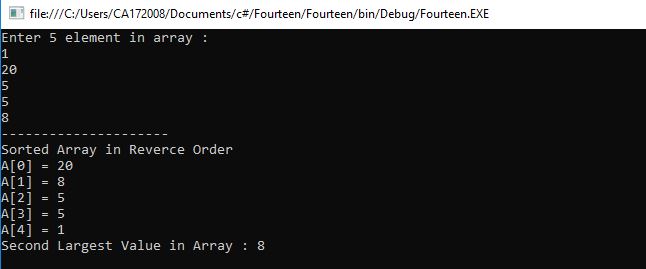
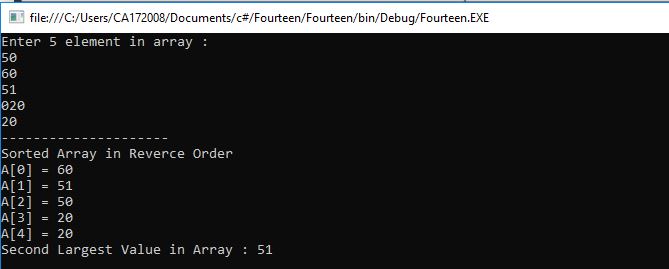
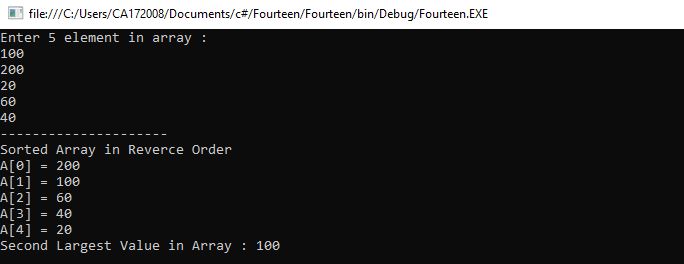
Console.ReadKey();

}

}

}

**OUTPUT**



1. **Program to illustrate the use of different properties in C#.**

using System;

using System.Collections.Generic;

using System.Text;

namespace Program

{

class PropertiesDemo

{

private string name;

private int age;

public string Name

{

set

{

name = value;

}

get

{

return name;

}

}

public int Age

{

set

{

if (value > 0)

age = value;

}

get

{

return age;

}

}

static void Main(string[] args)

{

PropertiesDemo p = new PropertiesDemo();

p.Name = "Vinayak";

p.Age = 23;

PropertiesDemo d = new PropertiesDemo();

d.Name = "Zutti";

d.Age = 22;

Console.WriteLine("{0} : {1}", p.Name, p.Age);

Console.WriteLine("{0} : {1}", d.Name, d.Age);

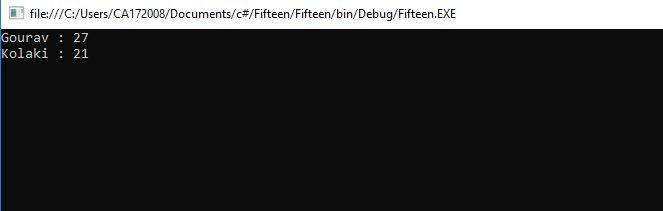
Console.ReadLine();

}

}

}

**OUTPUT**



1. **Demonstrate Use of Virtual and override keyword in C# with a**

**simple Program.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class A

{

public virtual void show()

{

Console.WriteLine("Hello: Base Class!");

Console.Write("\nPress Enter...");

Console.ReadLine();

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("---------------------------------------------------------");

Console.WriteLine("Use of Virtual and Overide Keyword.");

Console.WriteLine("---------------------------------------------------------");

Console.WriteLine("\nClass A is Base Class & Class B is derived from

A.\n");

Console.WriteLine("Creating Object of Class A.");

A a1 = new A();

a1.show();

Console.WriteLine("---------------------------------------------------------

\n");

Console.WriteLine("Creating Object of Class B.");

B b1 = new B();

b1.show();

Console.WriteLine("---------------------------------------------------------

\n");

Console.WriteLine("Creating Object of Class A and Calling Method of

Class

B.");

A a2 = new B();

a2.show();

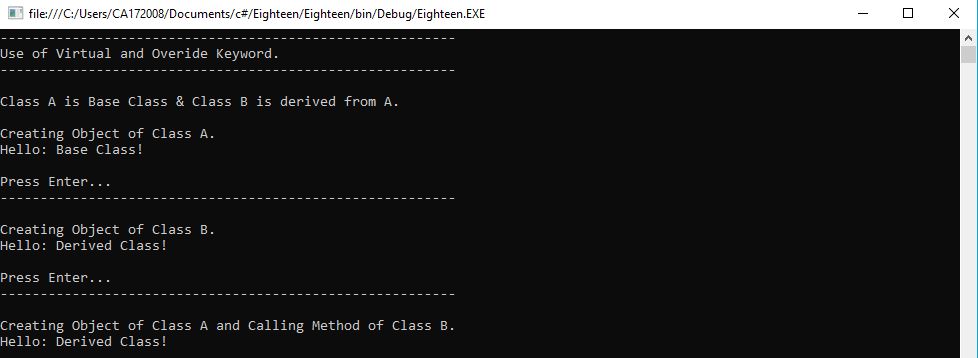
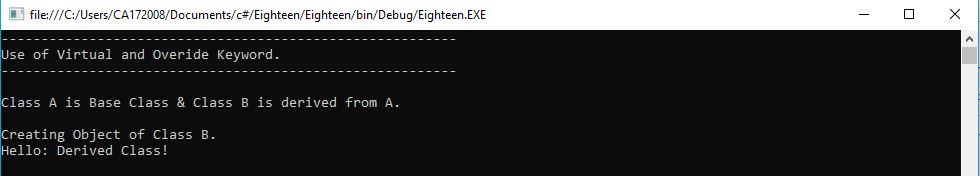
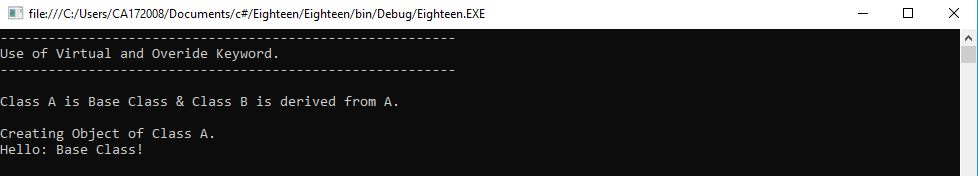
Console.ReadKey();

}

}

}

**OUTPUT**



1. **Program to multiply to matrices using Rectangular arrays.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication2

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("---------------------------------------------------------");

Console.WriteLine("Matrix Multiplication Using Rectanglular Array.");

Console.WriteLine("---------------------------------------------------------");

try

{

Console.WriteLine("Enter Rows and Column in 1st Matrix : ");

int r1 = Convert.ToInt16(Console.ReadLine());

int c1 = Convert.ToInt16(Console.ReadLine());

Console.WriteLine("Enter Rows and Column in 2nd Matrix : ");

int r2 = Convert.ToInt16(Console.ReadLine());

int c2 = Convert.ToInt16(Console.ReadLine());

if (r1 != c2)

{

Console.WriteLine("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("Matrix Multiplication Row Column Rule Violated.");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

else

{

int[,] mat1 = new int[r1, c1];

int[,] mat2 = new int[r2, c2];

int[,] mat3 = new int[r1, c2];

Console.WriteLine("Enter Element in Matrix one : ");

for (int i = 0; i < r1; i++)

{

for (int j = 0; j < c1; j++)

{

mat1[i, j] = (Convert.ToInt16(Console.ReadLine()));

}

}

Console.WriteLine("Enter Element in Matrix two : ");

for (int i = 0; i < r2; i++)

{

for (int j = 0; j < c2; j++)

{

mat2[i, j] = (Convert.ToInt16(Console.ReadLine()));

}

}

Console.WriteLine("\nFirst Matrix\n");

for (int i = 0; i < r1; i++)

{

for (int j = 0; j < c1; j++)

{

Console.Write("\t" + mat1[i, j]);

}

Console.WriteLine();

}

Console.WriteLine("\nSecond Matrix\n");

for (int i = 0; i < r2; i++)

{

for (int j = 0; j < c2; j++)

{

Console.Write("\t" + mat2[i, j]);

}

Console.WriteLine();

}

Console.WriteLine("\nMultiplication of Matrix\n");

for (int i = 0; i < r1; i++)

{

for (int j = 0; j < c2; j++)

{

for (int k = 0; k < c1; k++)

{

mat3[i, j] += mat1[i, k] \* mat2[k, j];

}

}

}

for (int i = 0; i < r2; i++)

{

for (int j = 0; j < c2; j++)

{

Console.Write("\t" + mat3[i, j]);

}

Console.WriteLine();

}

}

}

catch (Exception ex)

{

Console.WriteLine("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("Please Enter Numaric value.");

Console.WriteLine("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

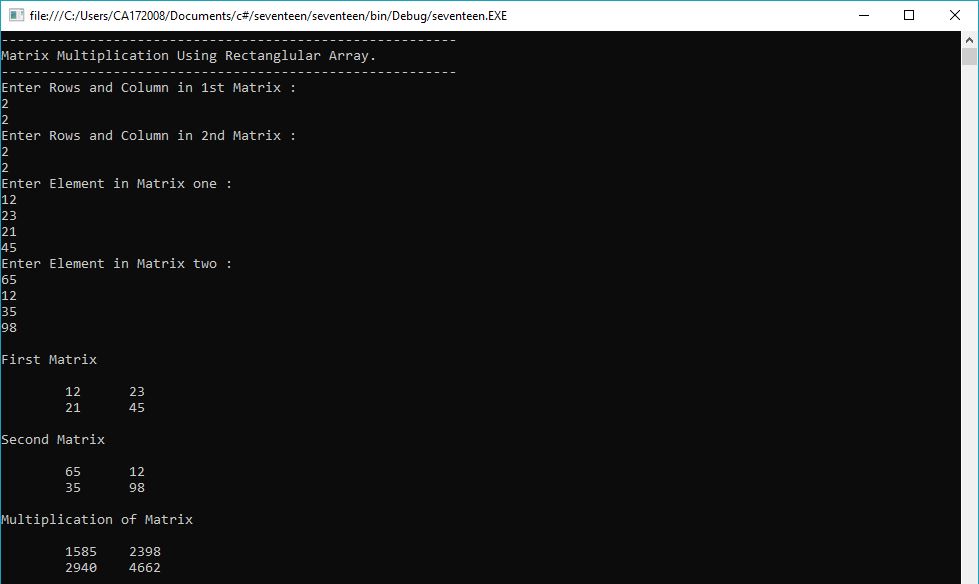
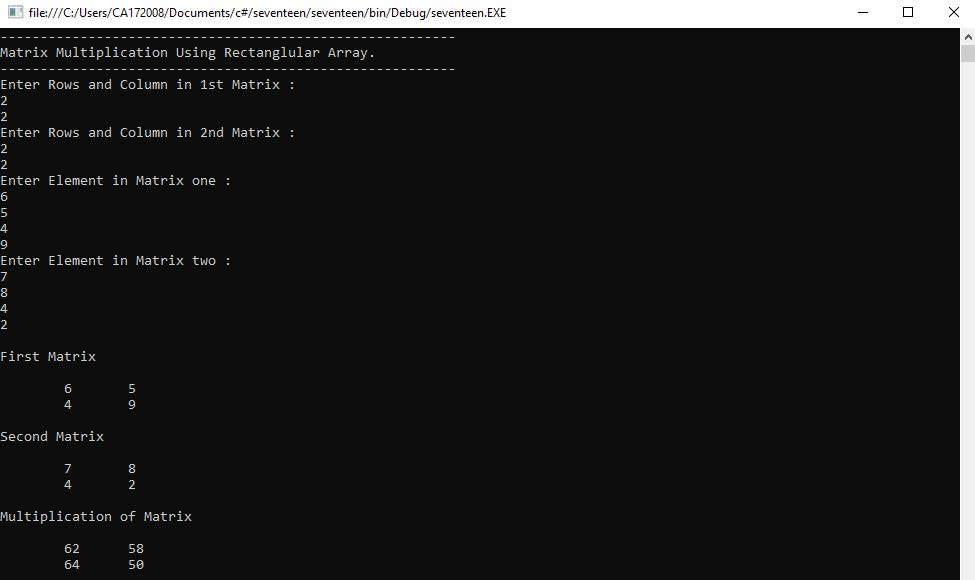
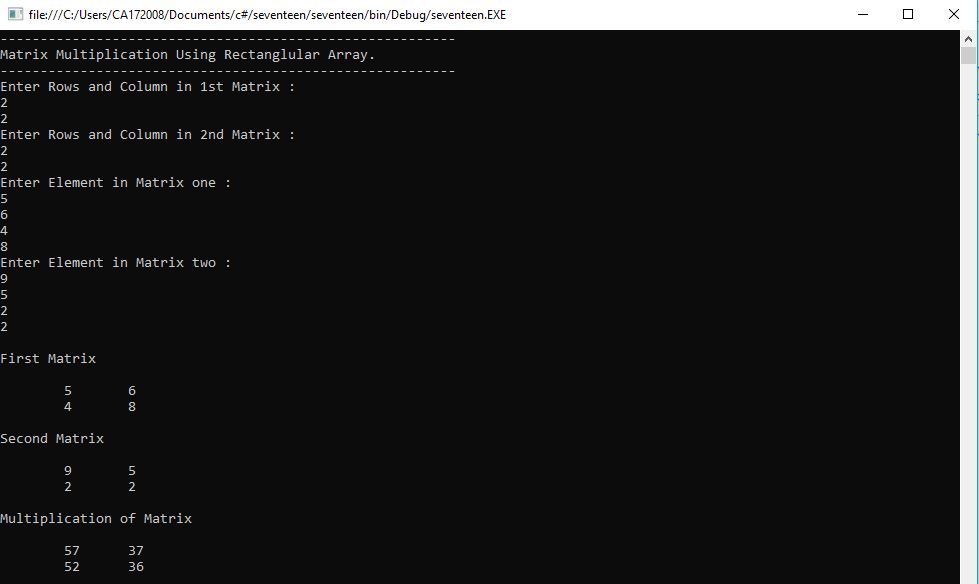
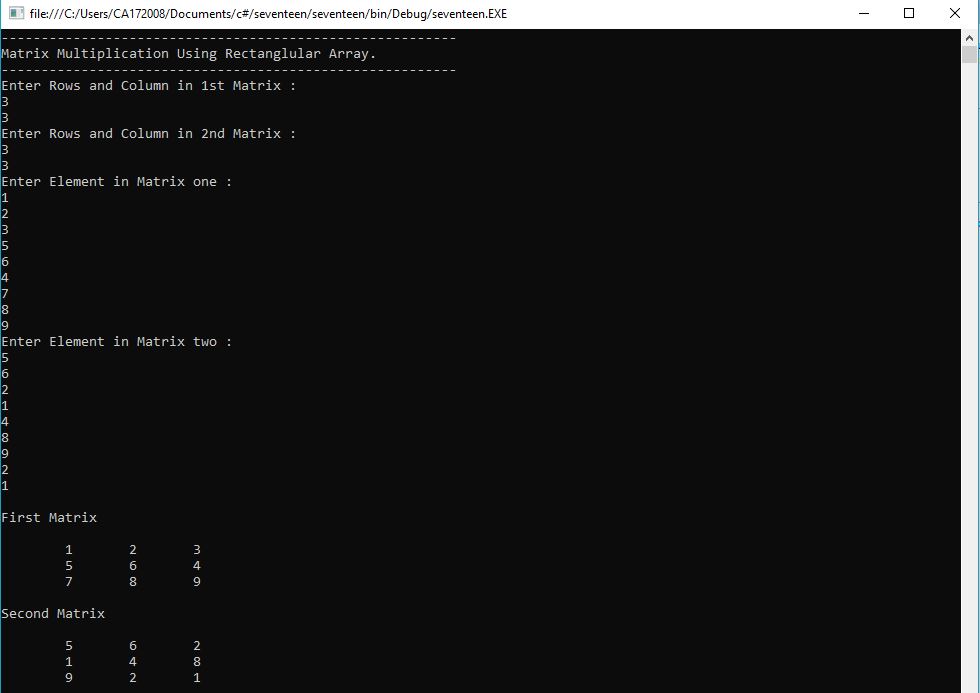
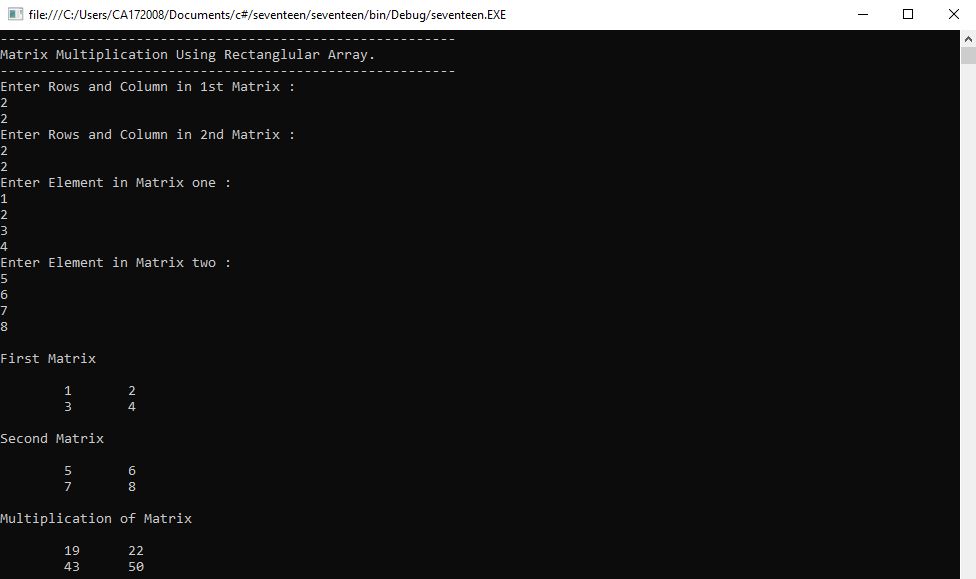
Console.ReadKey();

}

}

}

**OUTPUT**



1. **Perform operator overloading.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace OperatorOverloading

{

class Rectangle

{

int width;

int height;

Rectangle(int width, int height)

{

this.width = width;

this.height = height;

}

public static Rectangle operator +(Rectangle a, Rectangle b)

{

int totalWidth = a.width + b.width;

int totalHeight = a.height + b.height;

return new Rectangle(totalWidth, totalHeight);

}

static void Main(string[] args)

{

Rectangle r1 = new Rectangle(40, 60);

Rectangle r2 = new Rectangle(60, 40);

Console.WriteLine("-----------------");

Console.WriteLine("First Rectangle");

Console.WriteLine("-----------------");

Console.WriteLine("");

Console.WriteLine("Rectangle Width: {0}", r1.width);

Console.WriteLine("Rectangle Height: {0}", r1.height);

Console.WriteLine();

Console.WriteLine("-----------------");

Console.WriteLine("Second Rectangle");

Console.WriteLine("-----------------");

Console.WriteLine("");

Console.WriteLine("Rectangle Width: {0}", r2.width);

Console.WriteLine("Rectangle Height: {0}", r2.height);

Console.WriteLine();

Rectangle r3 = r1 + r2;

Console.WriteLine("Total Width: {0}", r3.width);

Console.WriteLine("Total Height: {0}", r3.height);

Console.ReadKey();

}

}

}

**OUTPUT**



1. **Create classes, they are reference types in C# and hence are allocated on the heap. Classes provide object-oriented constructs such as encapsulation, polymorphism, and inheritance. For instance, the program should print John. Doe twice, illustrating that objects are reference types, allocated on the heap implement the same using C#**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ProgramTwo

{

class Program

{

class User

{

private string name;

private string email;

public User(String name)

{

this.name = name;

}

public User(String name, String email)

{

this.name = name;

this.email = email;

}

public string getName()

{

return name;

}

public string getEmail()

{

return email;

}

public void setName(string name)

{

this.name = name;

}

public void setEmail(string email)

{

this.email = email;

}

}

class Admin : User

{

private string password;

public Admin(string name, string email, string password)

: base(name, email)

{

this.password = password;

}

public void setPassword(string password)

{

this.password = password;

}

public string getPassword()

{

return password;

}

static void Main(string[] args)

{

User user1 = new User("vinoo");

Admin user2 = new Admin("vinayak", "vinoo160496@gmail.com",

"vinu007");

Console.WriteLine("User 1:");

Console.WriteLine("Name: {0}", user1.getName());

Console.WriteLine("Email: {0}", user1.getEmail());

Console.WriteLine();

Console.WriteLine("User 2 (Admin):");

Console.WriteLine("Name: {0}", user2.getName());

Console.WriteLine("Email: {0}", user2.getEmail());

Console.WriteLine("Password: {0}", user2.getPassword());

Console.Read();

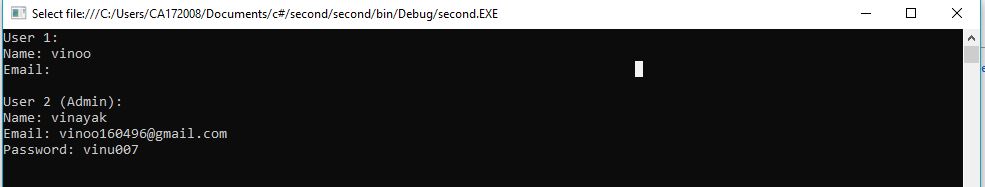
}

}

}

}

**OUTPUT**



1. **Work with Page using ASP.Net.**

**ASP.NET Page.**

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head id="Head1" runat="server">

<title>Game - Hit the button</title>

<style>

html{

height: 100%;

}

body{

min-height: 100%;

padding: 0;

margin: 0;

}

\*{

font-family: Arial;

}

.container{

height: 100vh;

display: flex;

flex-direction: column;

align-items:center;

}

h2{

font-size: 0.8em;

color: gray;

}

</style>

</head>

<body>

<form id="form1" runat="server">

<div class="container">

<h1>Welcome to Web Page - Hit the button!</h1>

<asp:Label ID="lblOutput" Text="Your score is 0" runat="server" />

<asp:button id="clickMeButton" runat="server" text="Hit!" onClick="clickMeButton\_Click" />

<br/>

<h2>&copy; 2019 Vinayak Z. All Rights Reserved.</h2>

</div>

</form>

</body>

</html>

**C# Page.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class \_Default : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void clickMeButton\_Click(object sender, EventArgs e)

{

object value = ViewState["HitCount"];

int i = (value == null) ? 1 : (int)value + 1;

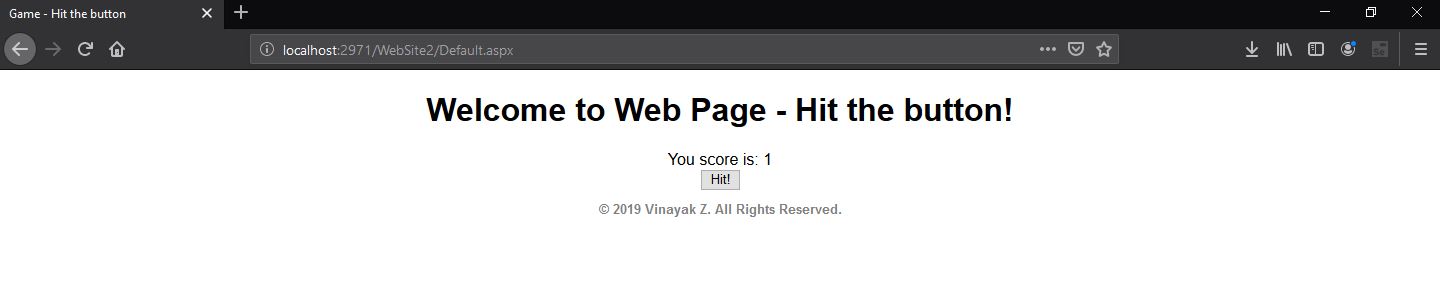
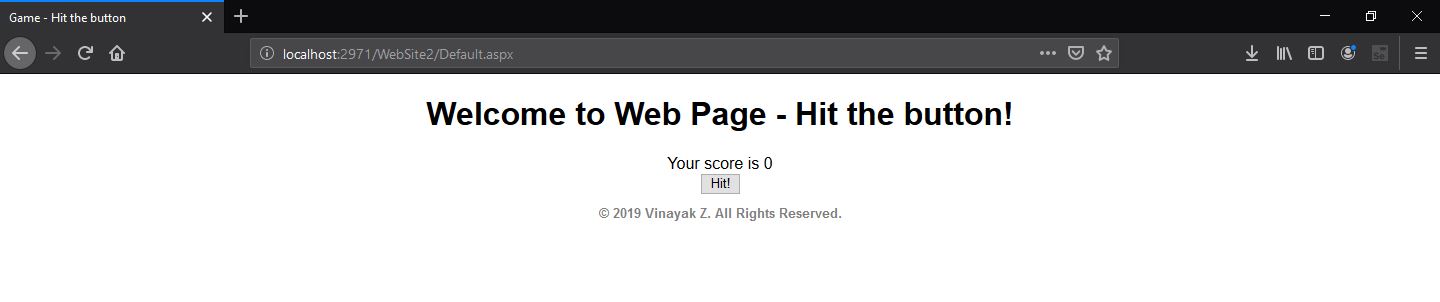
lblOutput.Text = string.Format("You score is: {0}", i);

ViewState["HitCount"] = i;

}

}

**OUTPUT**



1. **Work with forms using ASP.NET.**

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 WindowsFormsApplication2

{

public partial class Form1 : Form

{

string[] names;

string[] passs;

int rows;

public Form1()

{

InitializeComponent();

names = new string[10];

passs = new string[10];

names[0] = "admin";

names[1] = "user";

names[2] = "vinoo";

passs[0] = "admin";

passs[1] = "user";

passs[2] = "zutti";

rows = 3;

}

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

{

string username = textBox1.Text.Trim();

string password = textBox2.Text.Trim();

if (username.Equals("") || password.Equals(""))

{

MessageBox.Show("Fields cannot be empty!");

return;

}

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

{

if (names[i].Equals(username) && passs[i].Equals(password))

{

MessageBox.Show("Login Successfull!");

return;

}

}

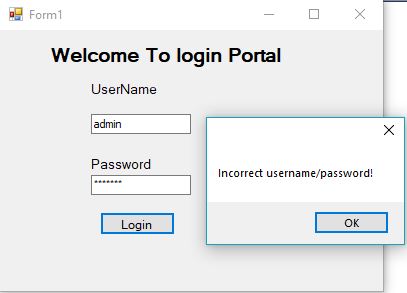
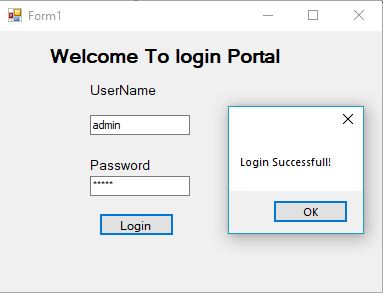
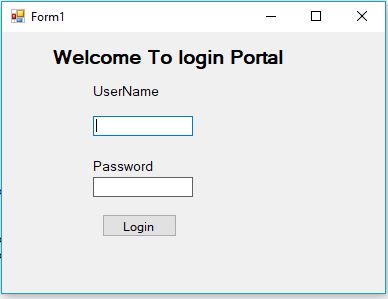
MessageBox.Show("Incorrect username/password!");

}

}

}

**OUTPUT**



1. **Describe Arrays and Strings methods with suitable C# program.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication5

{

class Program

{

static void Main(string[] args)

{

int[] array = { 1, 4, 6, 2, 8, 9, 7 };

Console.WriteLine("Properties & Methods of an Array: ");

displayArray(array);

Console.WriteLine();

Console.WriteLine("Length: {0}", array.Length);

Console.WriteLine("Rank: {0}", array.Rank);

Console.WriteLine("Max(): {0}", array.Max());

Console.WriteLine("Min(): {0}", array.Min());

Console.WriteLine("Sum(): {0}", array.Sum());

Console.WriteLine("Array.Reverse()");

Array.Reverse(array);

displayArray(array);

Console.WriteLine("Array.Sort()");

Array.Sort(array);

displayArray(array);

Console.WriteLine();

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.WriteLine();

Console.WriteLine("Properties & Methods of a String: ");

String str1 = "Hello World!, I am vinayak!. ";

Console.WriteLine();

String str2 = "Full Stack Developer.";

Console.WriteLine("String 1: {0}", str1);

Console.WriteLine("String 2: {0}", str2);

Console.WriteLine("str1.Length: {0}", str1.Length);

Console.WriteLine("str1.IndexOf('S'): {0}", str1.IndexOf('v'));

Console.WriteLine("str2.Contains() :{0}",

str2.Contains("Developer"));

Console.WriteLine("str1.Insert(19+6,\"-zutti\"):{0}",

str1.Insert(str1.IndexOf('v') + 6, "-zutti"));

Console.WriteLine("str1.Replace(\"I am\", \"This is\"): {0}",

str1.Replace("I am", "This is"));

Console.WriteLine("str1.Remove(str1.IndexOf(','):{0}",

str1.Remove(str1.IndexOf(',')));

Console.WriteLine("str1.Substring(str1.IndexOf(','):{0}",

str1.Substring(str1.IndexOf(',') + 1));

Console.WriteLine("String.Concat(str1, str2): {0}", String.Concat(str1, str2));

Console.WriteLine("String.Equals(str1, str2): {0}", String.Equals(str1, str2));

Console.WriteLine("String.Compare(str1, str2): {0}", String.Compare(str1, str2));

Console.ReadLine();

}

static void displayArray(int[] a)

{

Console.Write("[");

for (int i = 0; i < a.Length; i++)

{

Console.Write(" {0} ", a[i]);

}

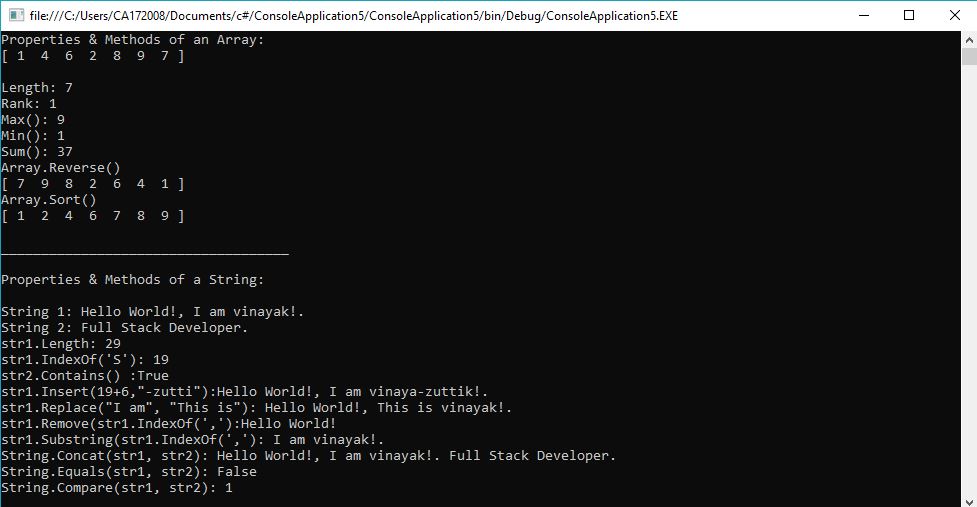
Console.WriteLine("]");

}

}

}

**OUTPUT**



1. **Describe delegates, events, errors and exceptions.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication6

{

class Car

{

public delegate void EventHandler(string msg);

public event EventHandler exploadListener;

public event EventHandler aboutToBlowListener;

private string name;

private bool isExhausted;

private int currentSpeed;

private const int maxSpeed = 140;

public Car(String name)

{

this.name = name;

}

public void accelerate(int delta)

{

if (isExhausted)

{

if (exploadListener != null)

exploadListener("Sorry, the car is dead!");

}

else

{

currentSpeed += delta;

if (10 >= maxSpeed - currentSpeed && aboutToBlowListener != null)

{

aboutToBlowListener("Be Careful, Gonna blow!");

}

if (currentSpeed >= maxSpeed)

isExhausted = true;

else

Console.WriteLine("-> Current Speed: {0}", currentSpeed);

}

}

}

class Program

{

static void Main(string[] args)

{

Car car = new Car("Tesla");

car.aboutToBlowListener += new Car.EventHandler(aboutToBlow);

car.exploadListener += new Car.EventHandler(exploded);

Console.WriteLine("\*\*\*\*\*Speeding Up\*\*\*\*\*\*\*");

try

{

for (int i = 0; i < 20; i++)

{

car.accelerate(20);

}

}

catch (Exception e)

{

Console.WriteLine("Exception: Car is dead already!");

}

Console.ReadLine();

}

public static void aboutToBlow(string msg)

{

Console.WriteLine("-> Reporting: {0}", msg);

}

public static void exploded(string msg)

{

Console.WriteLine("-> Reporting: {0}", msg);

throw new Exception("Car dead");

}

}

}

**OUTPUT**

