1. **Check Whether the Entered Year is a Leap Year or No.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace pgm3

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter Year : ");

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

if (((Year % 4 == 0) && (Year % 100 != 0)) || (Year % 400 == 0))

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

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

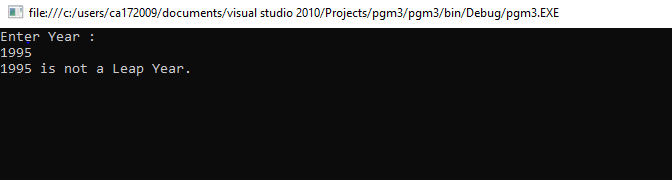
Console.ReadLine();

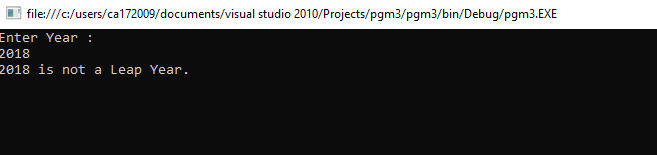
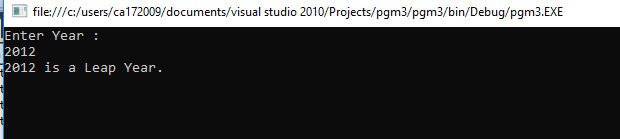
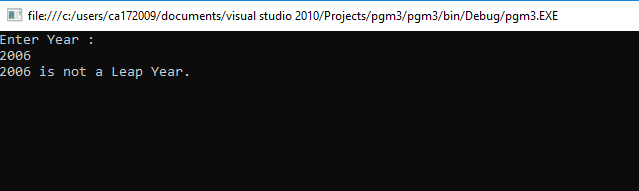
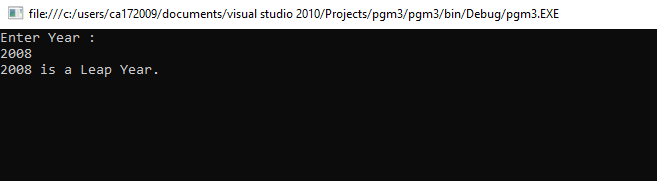
}

}

}

**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 pgm6

{

class Program

{

static void Main(string[] args)

{

int j, sum = 0;

Console.Write("The first 10 natural number are :\n");

for (j = 1; j <= 10; j++)

{

sum = sum + j;

Console.Write("{0} ",j);

Console.Write("\n");

}

Console.Write("\nThe Sum is : {0}\n", sum);

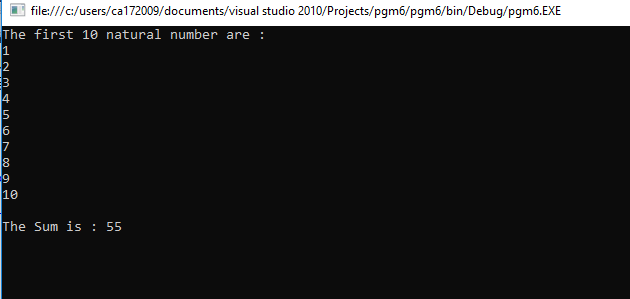
Console.ReadLine();

}

}

}

**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 ArthmaticOperation

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("This Program is developed by Shubham Sajannavar");

Console.WriteLine("Roll No : CA172007, Rani Channamma University, Belgavi");

int add, sub, mul, num1, num2;

float div;

Try

{

Console.WriteLine("Enter 1st Number : ");

num1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter 2nd Number : ");

num2 = Convert.ToInt32(Console.ReadLine());

add = num1 + num2;

sub = num1 - num2;

mul = num1 \* num2;

div = num1 / num2;

Console.WriteLine("Addition of\t\t"+num1+"and" + num2 + " = " + add);

Console.WriteLine("\nSubstration of \t\t"+num1+"and"+num2 + " = " + sub);

Console.WriteLine("Multiplication of \t"+num1+"and"+ num2 + " = " + mul);

Console.WriteLine("\nDivision of \t\t" +num1+"and"+num2 + " = " + div);

}

catch (Exception ex)

{

Console.WriteLine("Enter valid Number");

}

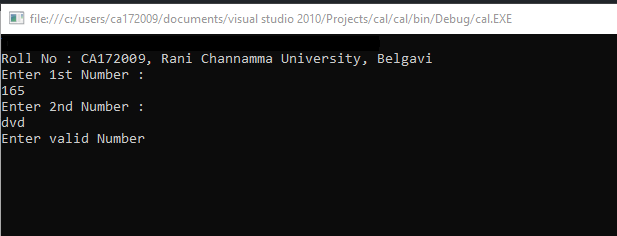
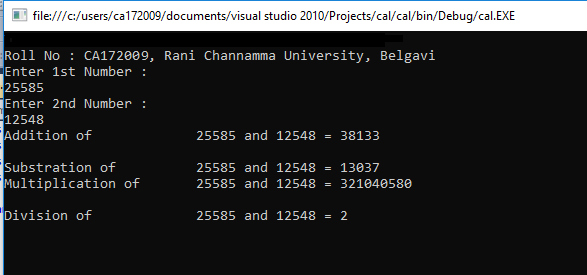
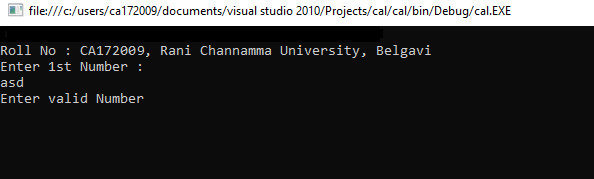
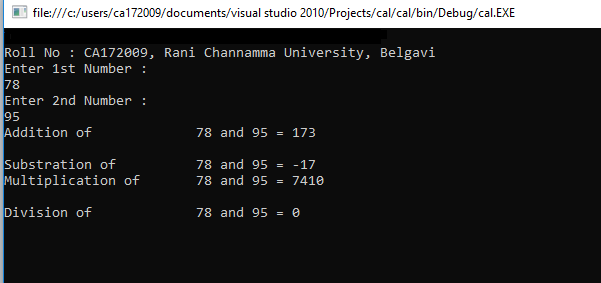
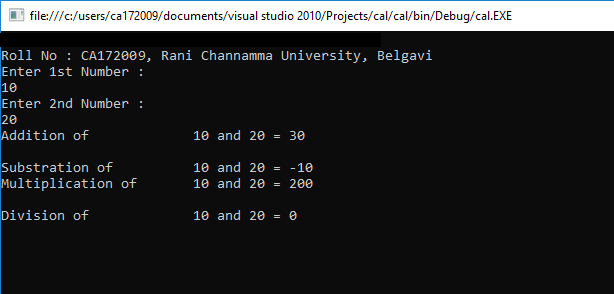
Console.ReadKey();

}

}

}

**OUTPUT**



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 CollegeDays

{

MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY

}

static void Main(string[] args)

{

foreach (var day in Enum.GetValues(typeof(CollegeDays)))

{

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

}

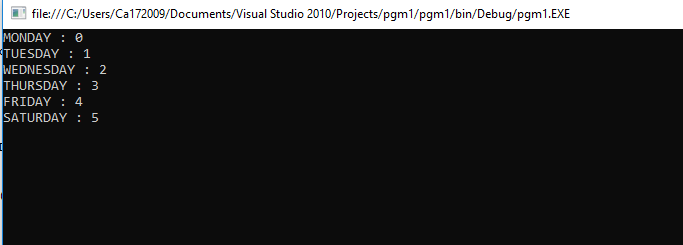
Console.Read();

}

}

}

**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 pgm7

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

float a;

float b;

float c;

a = Convert.ToInt32(textBox1.Text);

b = Convert.ToInt32(textBox2.Text);

c = a + b;

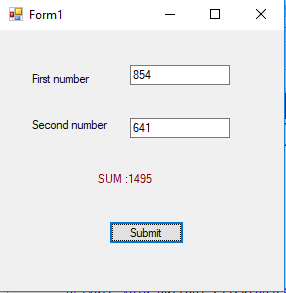
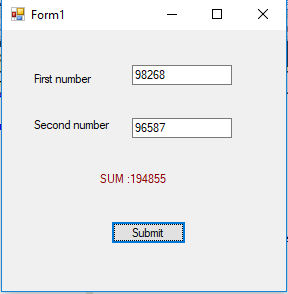
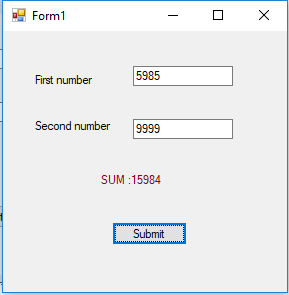
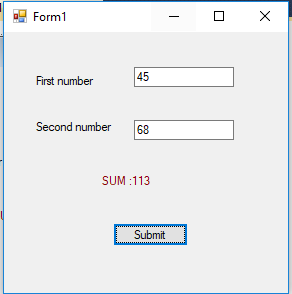
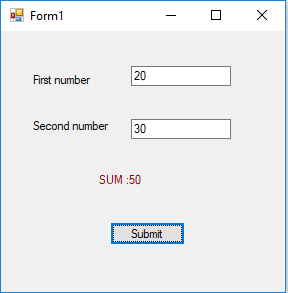
label3.Text = "SUM :" + c;

}

}

}

**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 LowUpp

{

public class Exercise15

{

public static void Main()

{

string str1;

char[] arr1;

int l, i;

l = 0;

char ch;

Console.Write("\n\nReplace lowercase characters by uppercase and vice-versa :\n");

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.Write("\n\n");

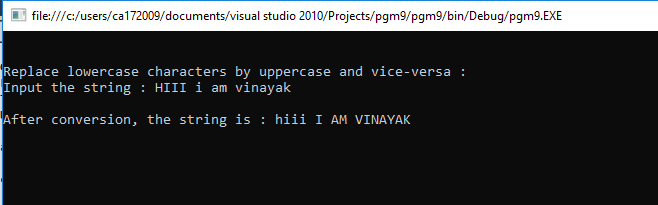
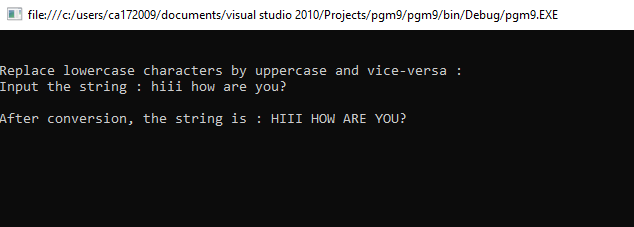
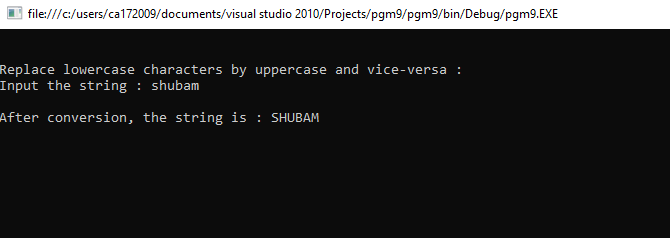
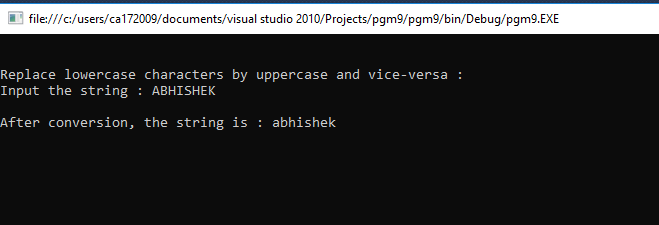
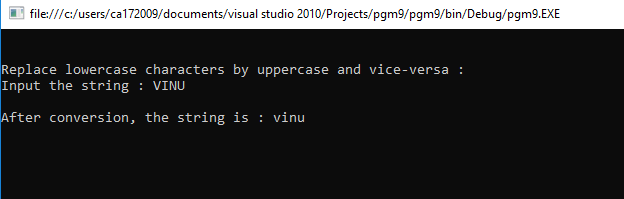
Console.ReadLine();

}

}

}

**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)

{

int[] arr = new int[5];

Console.WriteLine("Enter 5 array values");

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

{

//Console.WriteLine(i);

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

}

Array.Sort(arr);

Array.Reverse(arr);

Console.WriteLine("Second Highest Value In Array " + arr[1]);

foreach (var result in arr)

{

Console.Write(result + " ");

}

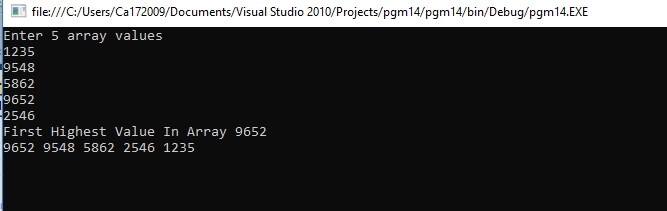
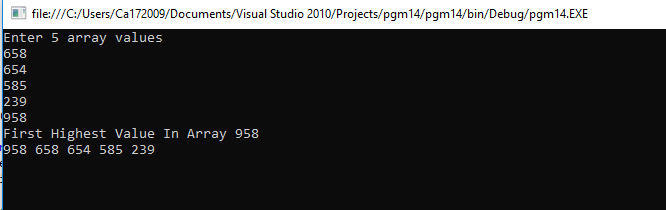
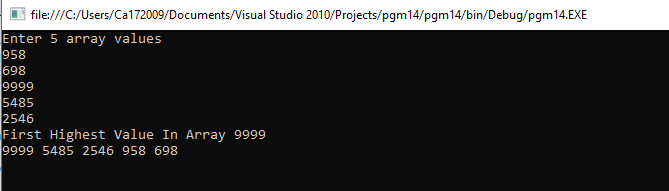
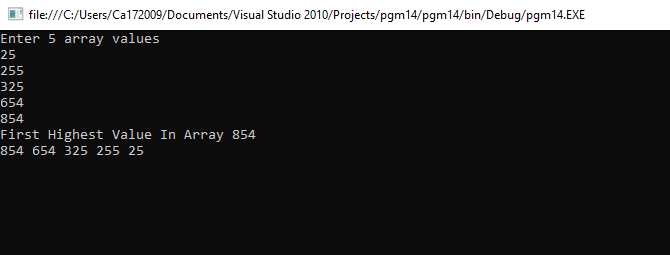
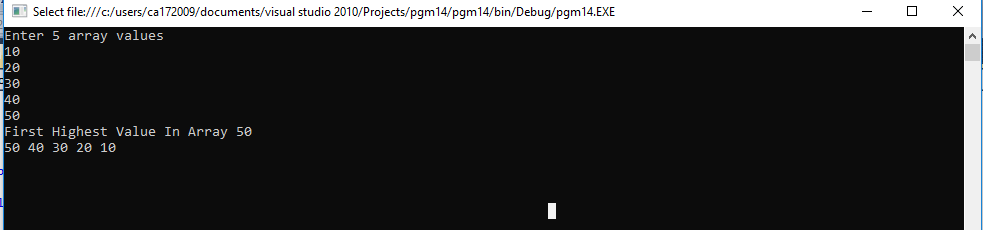
Console.ReadLine();

}

}

}

**OUTPUT**



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

using System;

namespace ProgramFifteen

{

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 = "Abhishek";

d.Age = -1;

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

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

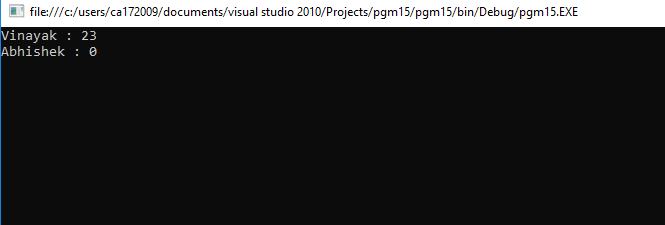
Console.ReadLine();

}

}

}

**OUTPUT**



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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace pgm16

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Argument length: " + args.Length);

Console.WriteLine("Given Arguments are:");

foreach (Object obj in args)

{

Console.WriteLine(obj);

}

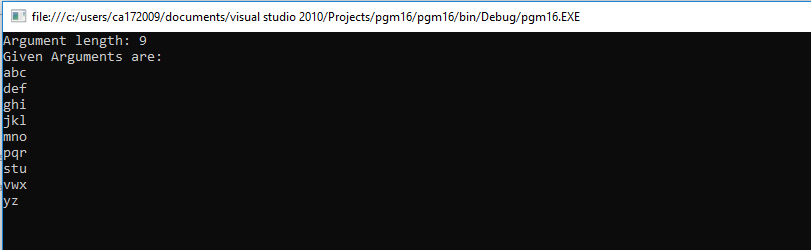
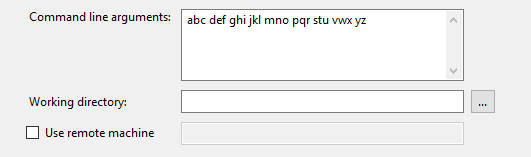
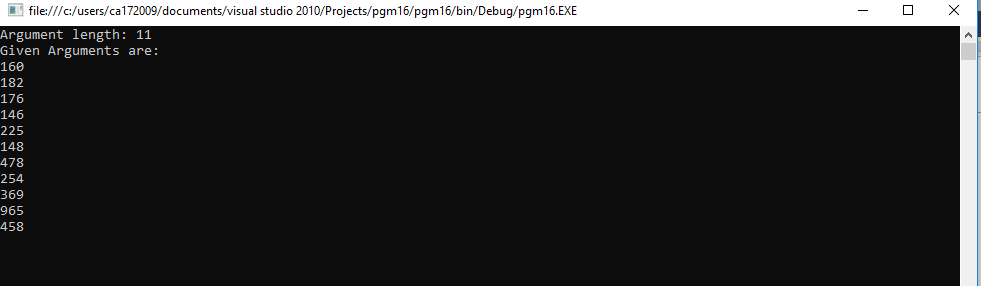
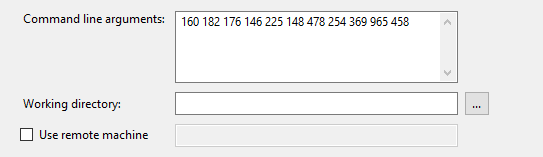
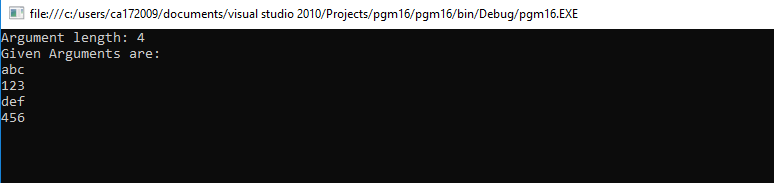
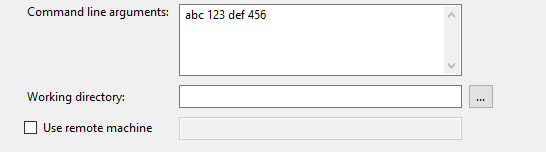
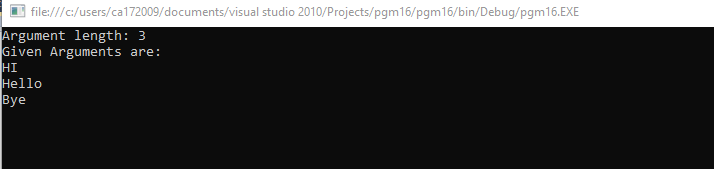
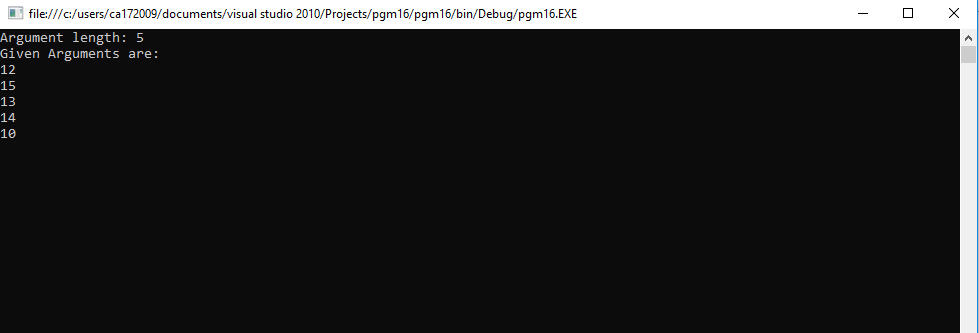
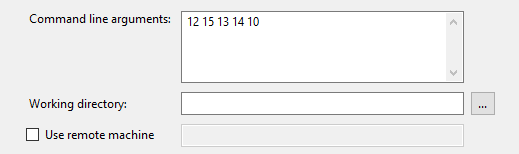
Console.ReadLine();

}

}

}

**OUTPUT**



1. **Create classes, they are reference type 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 pgmtow

{

class Program

{

static void Main(string[] args)

{

User user1 = new User("joy");

Admin user2 = new Admin("Vinayak", "Vinayak@gmail.com", "Kolaki");

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();

}

}

}

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;

}

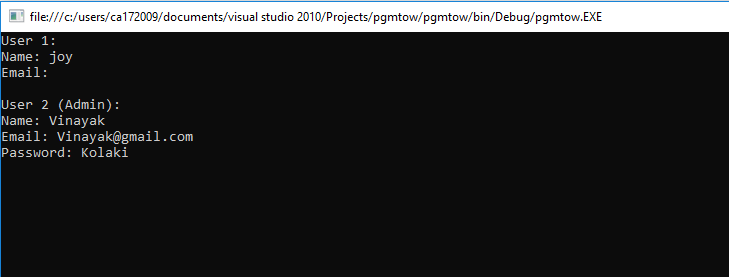
public string getPassword() {

return password;

}

}

**OUTPUT**



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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace pgm4

{

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("Properties & Methods of a String: ");

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

Console.WriteLine();

String str2 = "Full-Stack Android & Web Developer.";

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

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

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

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

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

Console.WriteLine("str1.Insert(19 + 6, \"-Kolaki\"): {0}", str1.Insert(str1.IndexOf('J') + 6, "-Kolaki"));

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]);

}

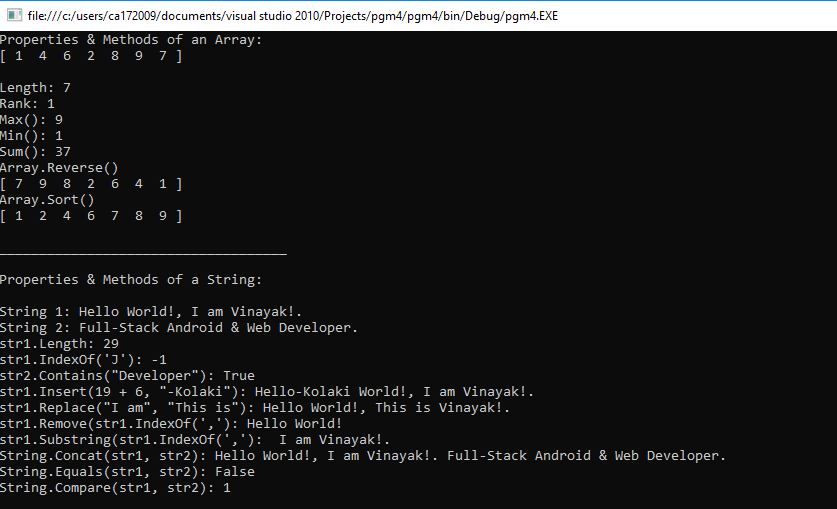
Console.WriteLine("]");

}

}

}

**OUTPUT**

****

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

**C#.net 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 Button1\_Click(object sender, EventArgs e)

{

object value = ViewState["HitCount"];

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

Label1.Text = string.Format("You score is: {0}", i);

ViewState["HitCount"] = i;

}

}

**ASP.net Page**

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

<head runat="server">

<title></title>

</head>

<body style="width: 625px; margin-left: 203px">

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

<div class="container">

<h1>Welcome to the page!</h1>

<br />

<asp:Label ID="Label1" Text="You clicked button 0 times" runat="server" />

<br />

<br />

<asp:button id="clickMeButton" runat="server" text="Click me"

onClick="Button1\_Click" />

<div class="space"> <br /> <footer>

<br />

<br />

Vinayak Kolaki(CA172009)</footer></div>

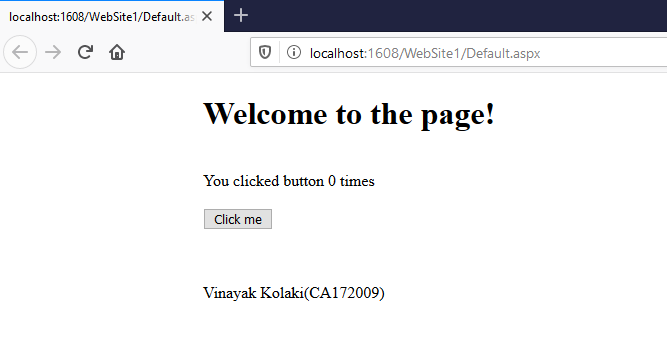
</div>

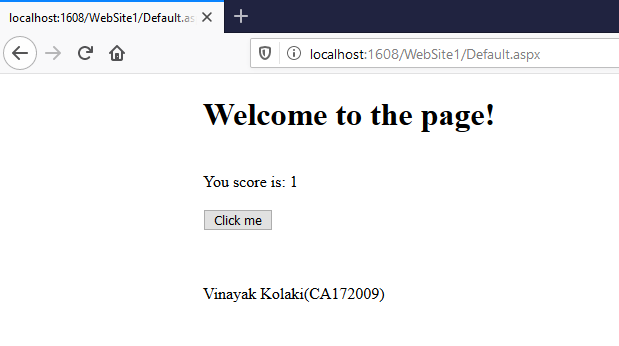
</form>

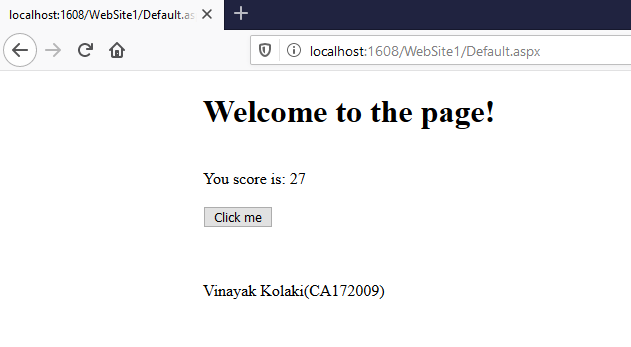
</body>

</html>

**Output**







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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace pgm13

{

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);

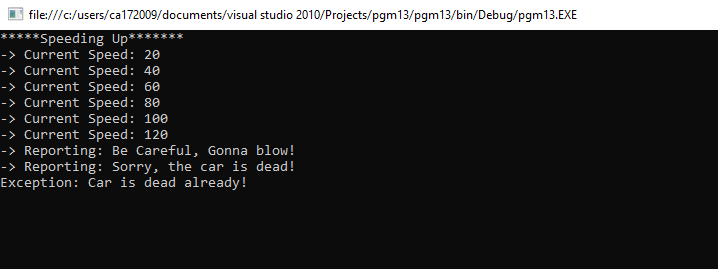
throw new Exception("Car dead");

}

}

}

**Output**

****

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 pgm1011

{

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] = "tony";

passs[0] = "admin";

passs[1] = "user";

passs[2] = "stark";

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;

}

}

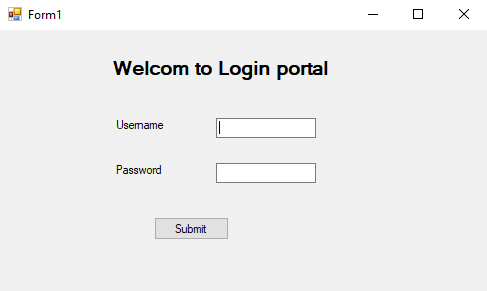
MessageBox.Show("Incorrect username/password!");

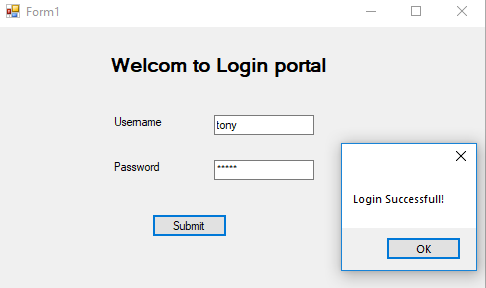
}

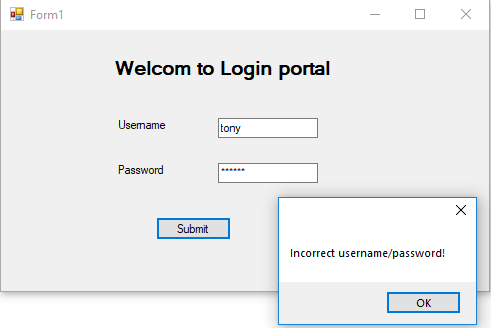
}

}

**Output**

****

****

****

1. **Perform Operator Overloading.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace operator\_overriding

{

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(95, 54);

Rectangle r2 = new Rectangle(53, 90);

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();

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

Console.WriteLine("Output");

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

Console.WriteLine("");

Rectangle r3 = r1 + r2;

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

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

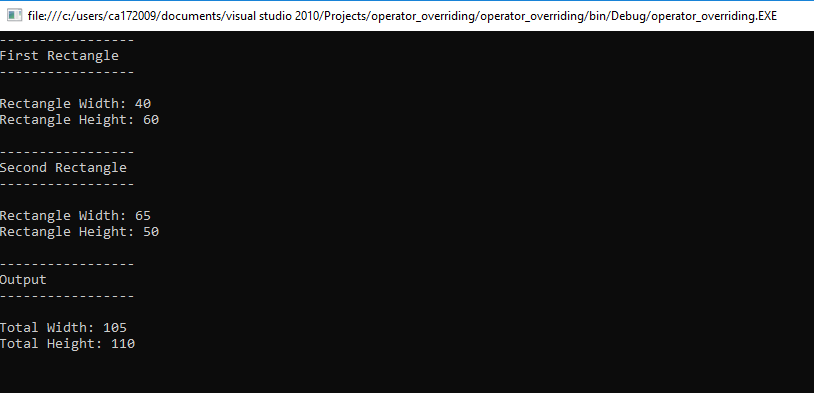
Console.ReadKey();

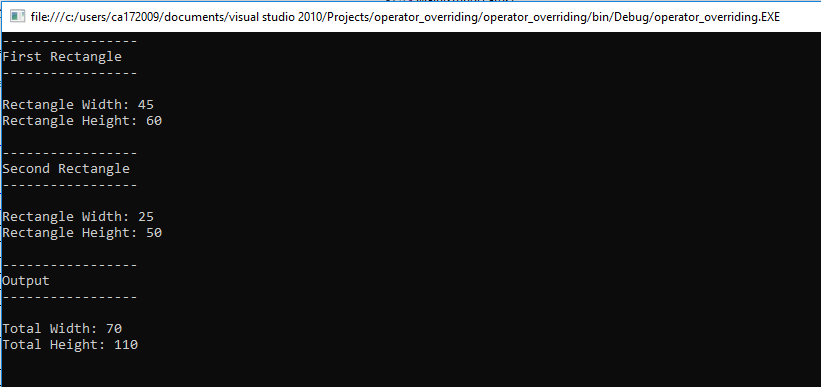
}

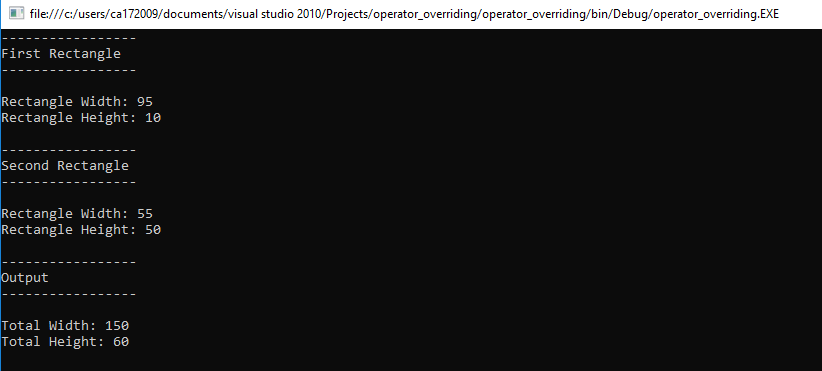
}

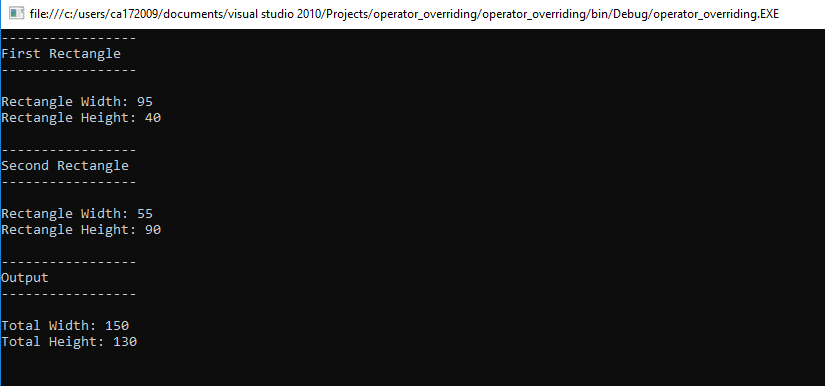
}

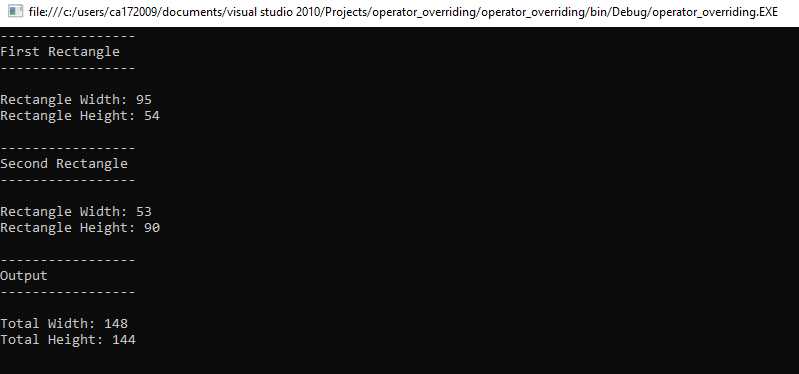
**Output**

****

****

****

****

****

1. **Program to Multiply to matrices using Rectangle arrays.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Matricies\_Multiplication

{

class Program

{

static void Main(string[] args)

{

Console.Title = ("Matix Multiplication");

int[,] mat1 = new int[2, 2];

int[,] mat2 = new int[2, 2];

int[,] mat3 = new int[2, 2];

Console.WriteLine("Enter Element for 1st Array ");

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < 2; j++)

{

mat1[i, j] = Convert.ToInt32(Console.ReadLine());

}

}

Console.WriteLine("Matrix one element are stored.\n");

Console.WriteLine("Enter Element for 2nd Array.");

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < 2; j++)

{

mat2[i, j] = Convert.ToInt32(Console.ReadLine());

}

}

int r1 = mat1.GetLength(0);

int c1 = mat1.GetLength(1);

int r2 = mat2.GetLength(0);

int c2 = mat1.GetLength(1);

//Console.WriteLine("\n\t\tRows \tColumn");

//Console.WriteLine("Matrix 1\t" + r1.ToString() + "\t" + c1.ToString());

//Console.WriteLine("Matrix 2\t" + r2.ToString() + "\t" + c2.ToString());

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

Console.WriteLine("Matrix two element are stored.");

Console.WriteLine("First Array");

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < 2; j++)

{

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

}

Console.WriteLine();

}

Console.WriteLine("Second Array");

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < 2; j++)

{

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

}

Console.WriteLine();

}

Console.WriteLine("\n");

Console.WriteLine("Multiplication of two matrix");

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 < 2; i++)

{

for (int j = 0; j < 2; j++)

{

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

}

Console.WriteLine();

}

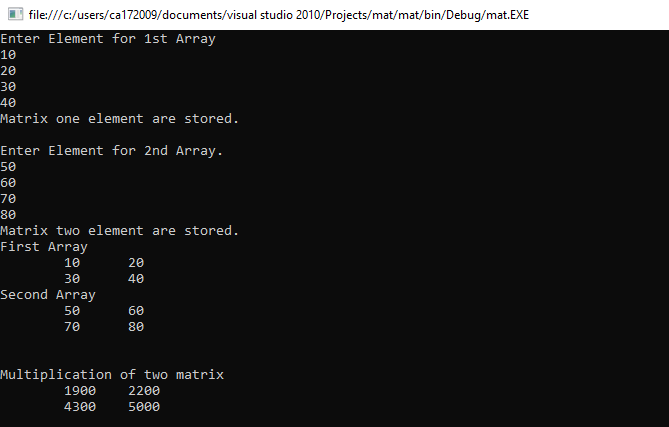
Console.ReadKey();

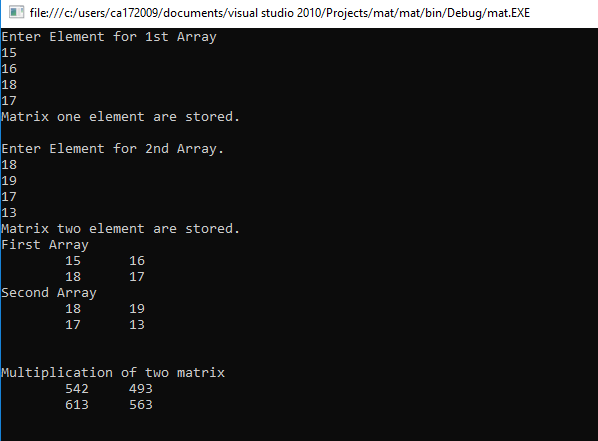
}

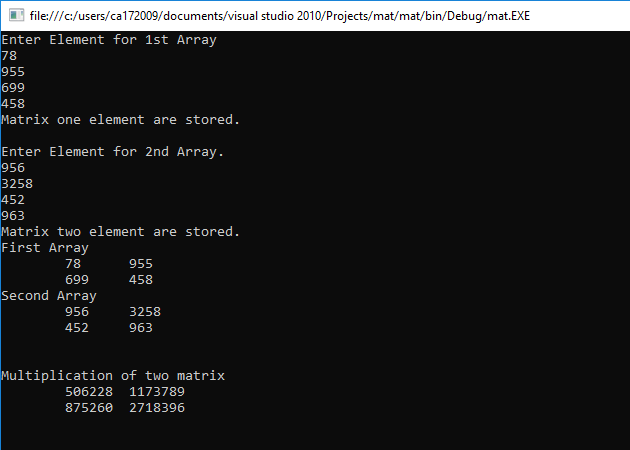
}

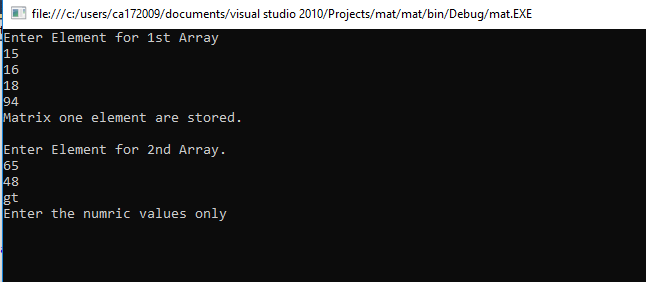
}

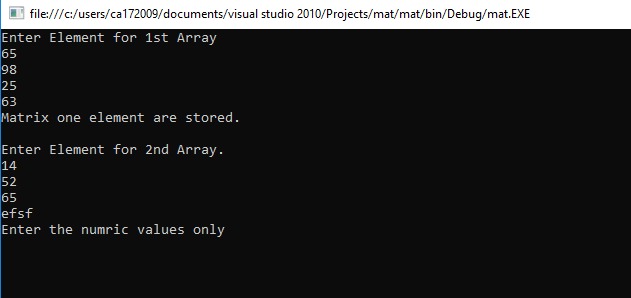
**Output**











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 va

{

class A

{

public virtual void show()

{

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

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

Console.ReadLine();

}

}

class B : A

{

public override void show()

{

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

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

Console.ReadLine();

}

}

class Program

{

static void Main(string[] args)

{

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();

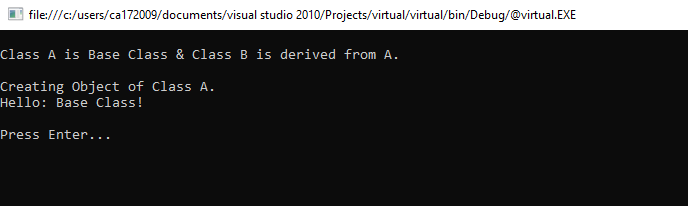
Console.ReadKey();

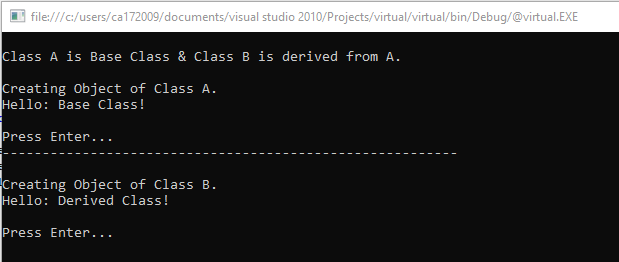
}

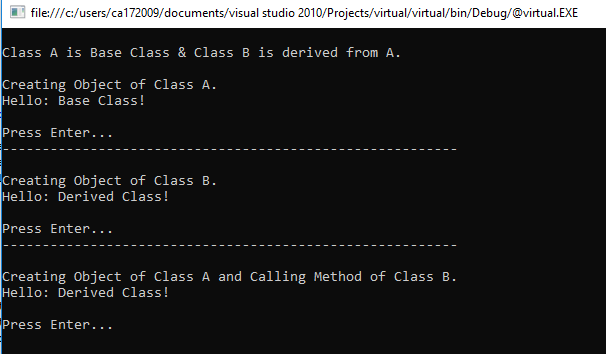
}

}

**Output**

****

****

****

1. **Describe access data source through ADO.NET.**

**Form.cs**

using System;

using System.Collections.Generic;

using System.Data;

using System.Windows.Forms;

namespace ProgramEleven

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

UserAccessLayer uAL = new UserAccessLayer();

List<User> users = uAL.getAllUsers();

if(users.Count == 0)

lblStatus.Text = "No data!";

else

lblStatus.Text = "Data Fetched!";

dGV.DataSource = users;

}

}

}

**Users.cs**

using System;

namespace ProgramEleven

{

class User

{

public int Id

{

get;

set;

}

public string UserName

{

get;

set;

}

public string RollNumber

{

get;

set;

}

public string Email

{

get;

set;

}

}

}

**UserAccessLayer.cs**

using System;

using System.Data;

using System.Data.SqlClient;

namespace ProgramEleven

{

class UserAccessLayer

{

private List<User> users;

private string connectionString = @"Data Source=.\SQLEXPRESS/PSELF;Initial

Catalog=TestDB; Integrated Security=True";

private SqlConnection connection;

private SqlCommand command;

private string query;

public List<User> getAllUsers()

{

users = new List<User>();

try

{

connection = new SqlConnection(connectionString);

connection.Open();

query = "SELECT \* FROM user";

command = new SqlCommand(query, connection);

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

User user = new User();

user.Id = Convert.ToInt16(reader.GetValue(0));

user.UserName = reader.GetValue(1).ToString();

user.Email = reader.GetValue(2).ToString();

user.RollNumber = reader.GetValue(3).ToString();

users.Add(user);

}

}

catch (SqlException ex)

{

Console.WriteLine("Error in fetching database!: " + ex.Message);

}

return users;

}

}

}

**Output**

