

Visual Programming Languages

Lab Manual

[Fall 2019]

|  |  |
| --- | --- |
| Student Name: Muhammad Ather Anwer |  |
| Student Id: 12403 |  |
|  |  |

|  |  |
| --- | --- |
| Prepared By: *Dr. Noman Islam* |  |
| Instructor: *Dr. Noman Islam* |  |

**LIST OF EXPERIMENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Date** | **Experiment** |  |
| **1** | 3/009/19 | To setup the environment and familiarize with C# |  |
| **2** | 10/09/19 | To study and understand how to write programs in C# using loops and arrays |  |
| **3** | 27/09/19 | To study and implement object oriented programming concepts in C# |  |
| **4** | 17/10/19 | To study and implement Windows Forms application in C# |  |
| **5** | 24/09/19 | To study and implement Collections in C# |  |
| **6** | 7/10/19 | To study and implement I/O in C# |  |
| **7** | 14/10/19 | To study and implement XML parsing in C# |  |
| **8** | 21/11/19 | To study and implement WPF and its layouts in C# |  |
| **9**  **10** | 2811/19  17/12/19 | To study and implement LINQ in C#  To study and Implement ADO in C# |  |
|  |  |  |  |

**Lab 1: To setup the environment and familiarize with C#**

The objective of this lab is to set up the Visual Studio environment and get some familiarity with the C# language.

Download and install Visual Studio .Net. Visual Studio is the leading platform powered by Microsoft for development on .net framework

**Lab Tasks:**

1. Write a small program in C# to print your CV.

TASK 1 Solution (CODE)

using System;

namespace ConsoleApp3

{

class Program

{

static void Main(string[] args)

{

//Task1

Console.WriteLine("Muhammad Ather Anwer");

Console.WriteLine("Anwer Saeed Yousufi");

Console.WriteLine("22");

Console.WriteLine("mhmmdather@gmail.com");

Console.WriteLine("+9233535234");

Console.WriteLine("House A-436");

Console.WriteLine("Karachi");

Console.WriteLine("Sindh");

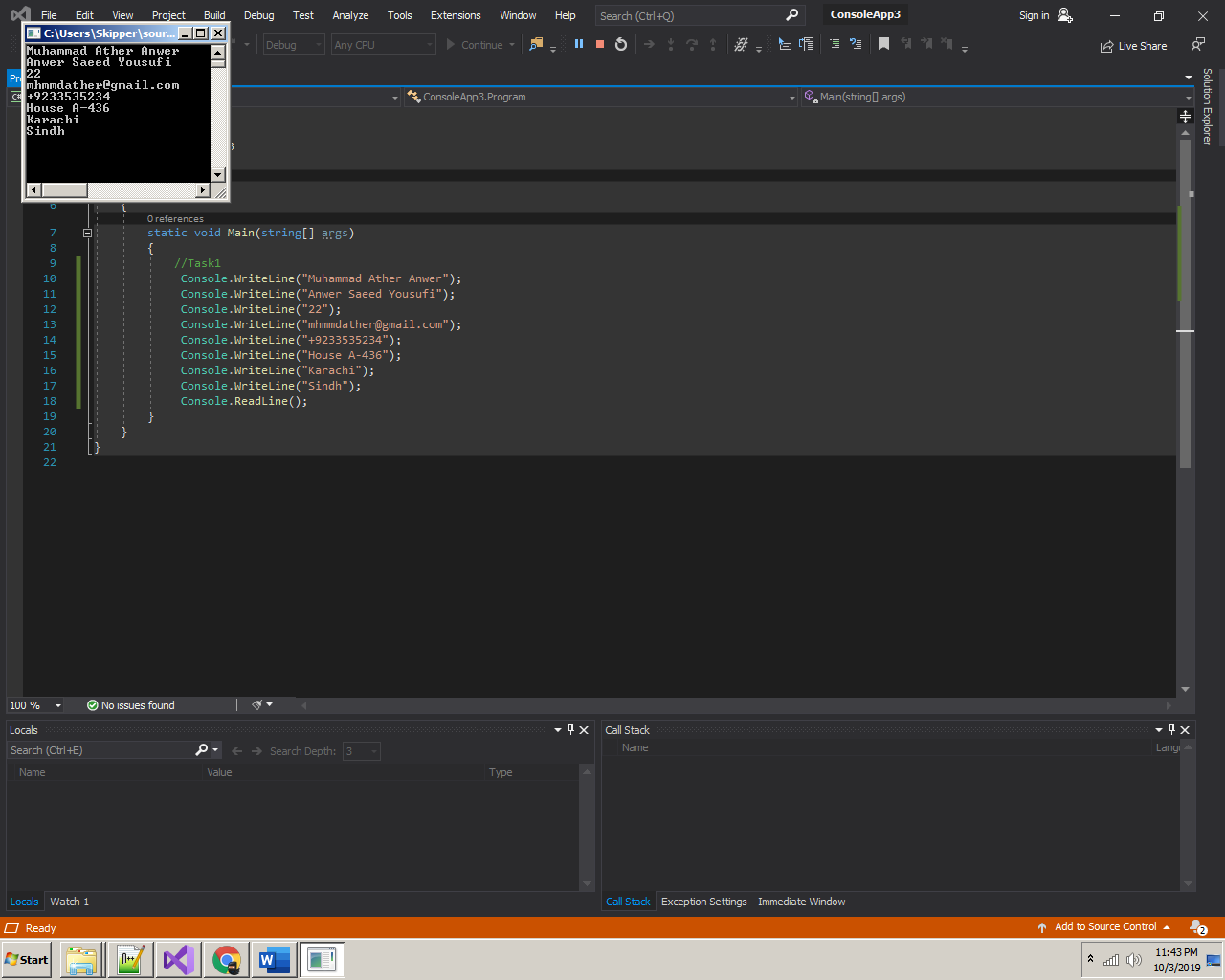
Console.ReadLine();

}

}

}

OUTPUT



1. Write a program to calculate whether an input number is even or odd.

TASK 2 Solution (CODE)

using System;

namespace ConsoleApp3

{

class Program

{

static void Main(string[] args)

{

//Task 2

int a;

Console.WriteLine("Enter a number to find if its even or odd: ");

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

if (a % 2 == 0)

{

Console.WriteLine(a + "Is even number ");

Console.Read();

}

Console.WriteLine(a + "Is odd number ");

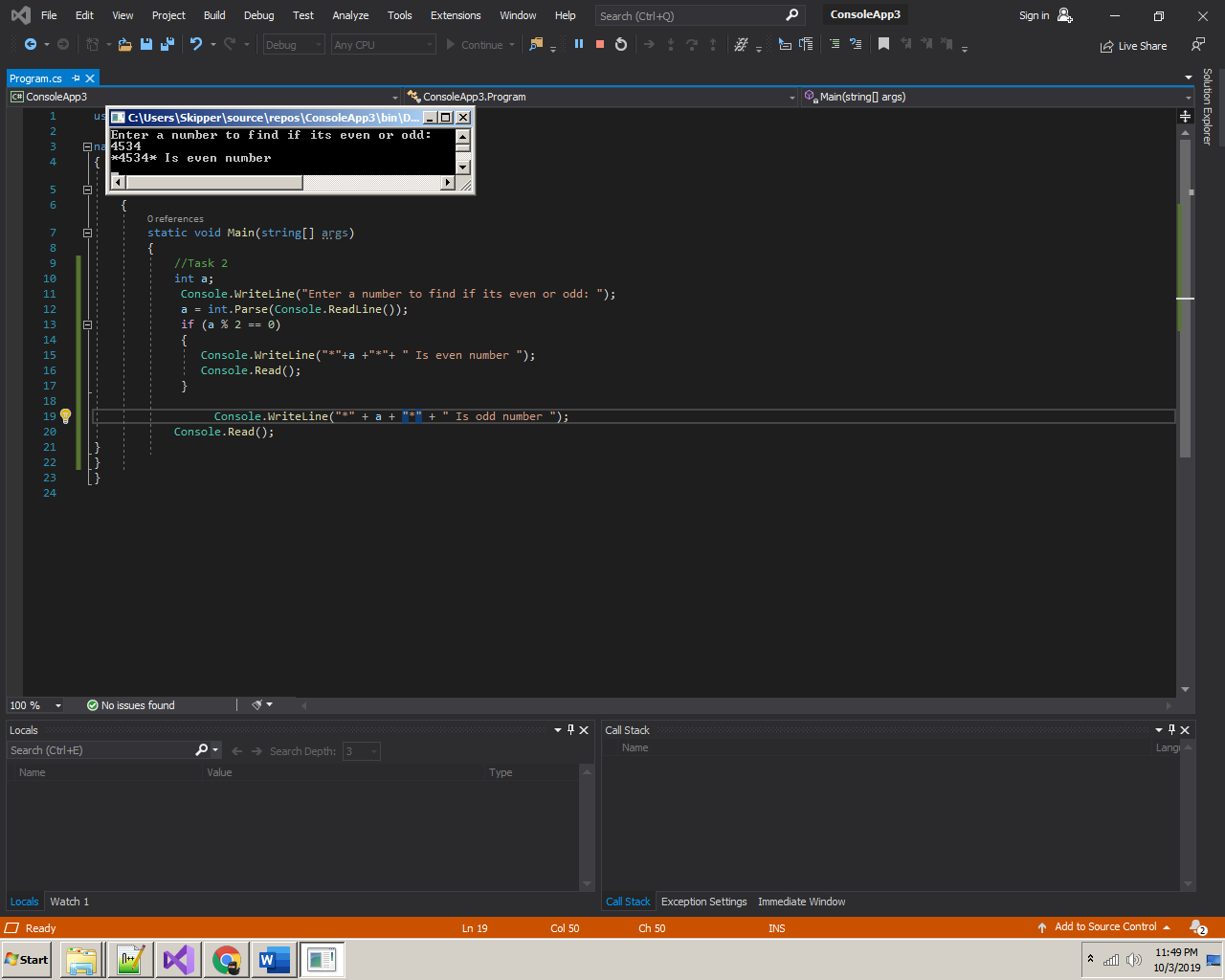
Console.Read();

}

}

}

OUTPUT



1. Write a program that takes thee numbers from user as input. The program then prints the maximum and minimum of the input numbers.

TASK 3 Solution (CODE)

using System;

namespace ConsoleApp3

{

class Program

{

static void Main(string[] args)

{

// Task 3

Console.WriteLine("\nInput first integer:");

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

Console.WriteLine("Input second integer:");

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

Console.WriteLine("Input third integer:");

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

Console.WriteLine("Largest of three: " + Math.Max(x, Math.Max(y, z)));

Console.WriteLine("Lowest of three: " + Math.Min(x, Math.Min(y, z)));

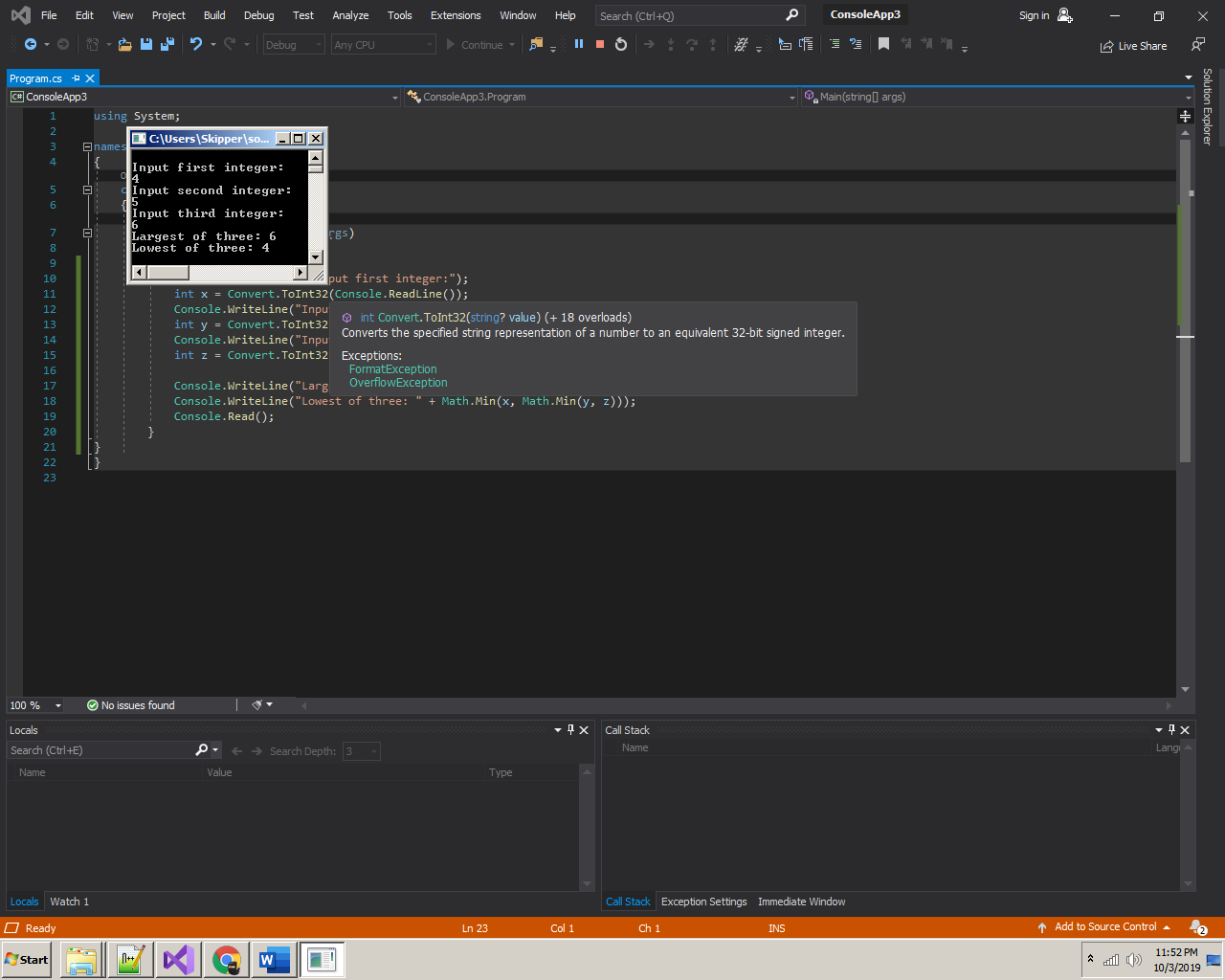
Console.Read();

}

}

}

OUTPUT



1. Write a program that takes the month (1…12) as input. Print whether the season is summer, winter, spring or autumn depending upon the input month.

TASK 4 Solution (CODE)

using System;

namespace ConsoleApp3

{

class Program

{

static void Main(string[] args)

{

//Task 4

int monno;

Console.Write("\n\n");

Console.Write("Read month number and display month name:\n");

Console.Write("-------------------------------------------");

Console.Write("\n\n");

Console.Write("Input Month No : ");

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

switch (monno)

{

case 1:

Console.Write("January\n");

Console.Write("Season is winter\n");

Console.Read();

break;

case 2:

Console.Write("February\n");

Console.Write("Season is winter\n");

Console.Read();

break;

case 3:

Console.Write("March\n");

Console.Write("Season is spring\n");

Console.Read();

break;

case 4:

Console.Write("April\n");

Console.Write("Season is spring\n");

Console.Read();

break;

case 5:

Console.Write("May\n");

Console.Write("Season is spring\n");

Console.Read();

break;

case 6:

Console.Write("June\n");

Console.Write("Season is summer\n");

Console.Read();

break;

case 7:

Console.Write("July\n");

Console.Write("Season is summer\n");

Console.Read();

break;

case 8:

Console.Write("August\n");

Console.Write("Season is summer\n");

Console.Read();

break;

case 9:

Console.Write("September\n");

Console.Write("Season is fall\n");

Console.Read();

break;

case 10:

Console.Write("October\n");

Console.Write("Season is fall\n");

Console.Read();

break;

case 11:

Console.Write("November\n");

Console.Write("Season is fall\n");

Console.Read();

break;

case 12:

Console.Write("December\n");

Console.Write("Season is winter\n");

Console.Read();

break;

default:

Console.Write("invalid Month number. \nPlease try again ....\n");

Console.Read();

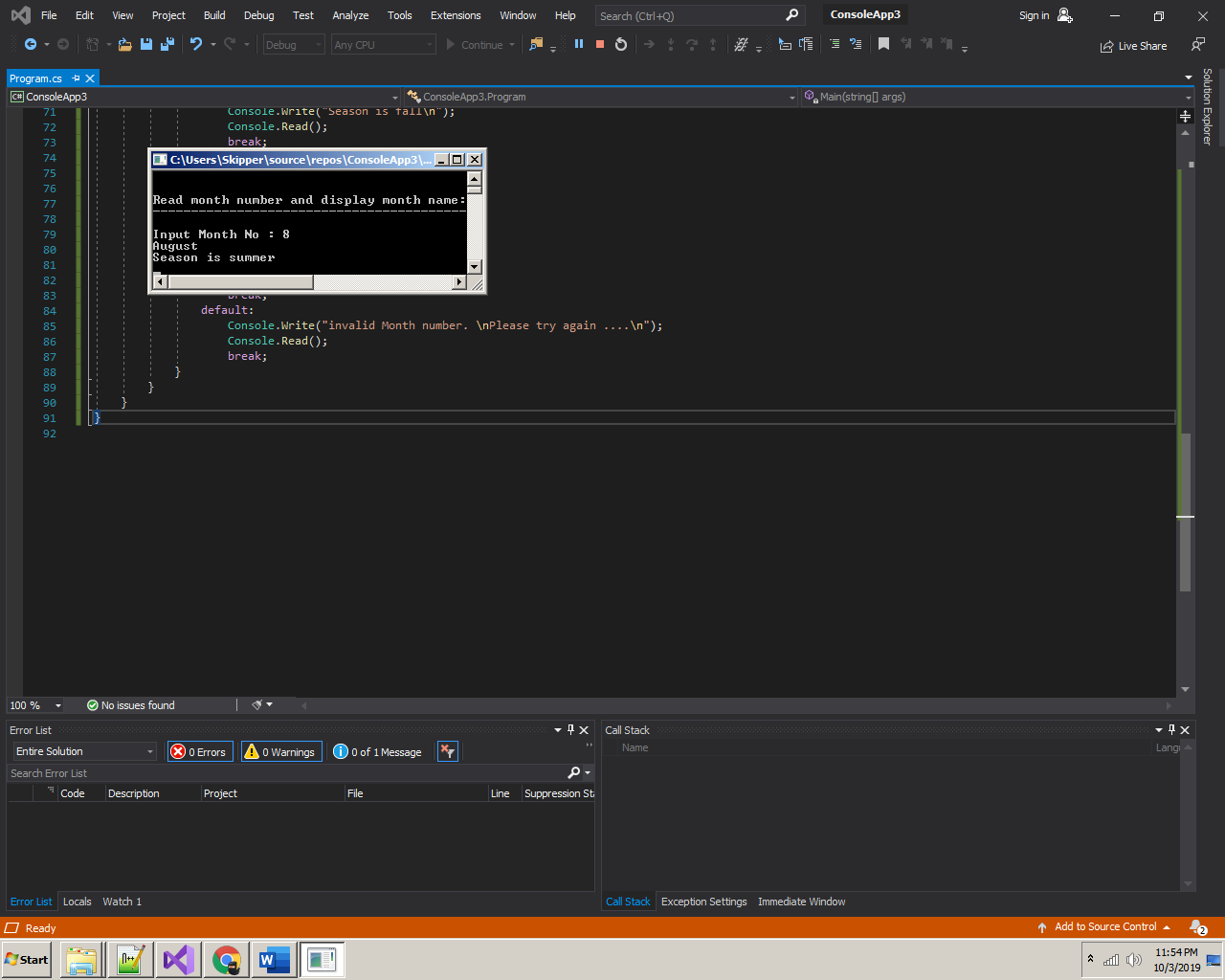
break;

}

}

}

OUTPUT



1. To determine whether a year is a leap year, follow these steps:
   1. If the year is evenly divisible by 4, go to step 2. Otherwise, go to step 5.
   2. If the year is evenly divisible by 100, go to step 3. Otherwise, go to step 4.
   3. If the year is evenly divisible by 400, go to step 4. Otherwise, go to step 5.
   4. The year is a leap year (it has 366 days).
   5. The year is not a leap year (it has 365 days).

Write a program to input an year as integer. Using if…else, determines whether the input is a leap year or not.

TASK 5 Solution (CODE)

using System;

namespace ConsoleApp3

{

class Program

{

static void Main(string[] args)

{

//Task 5

Console.Write("\n\n");

Console.Write("Check whether a given year is leap year or not:\n");

Console.Write("----------------------------------------------");

Console.Write("\n\n");

Console.Write("Input an year : ");

int chk\_year = Convert.ToInt32(Console.ReadLine());

if ((chk\_year % 400) == 0)

Console.WriteLine("{0} is a leap year.\n", chk\_year);

else if ((chk\_year % 100) == 0)

Console.WriteLine("{0} is not a leap year.\n", chk\_year);

else if ((chk\_year % 4) == 0)

Console.WriteLine("{0} is a leap year.\n", chk\_year);

else

Console.WriteLine("{0} is not a leap year.\n", chk\_year);

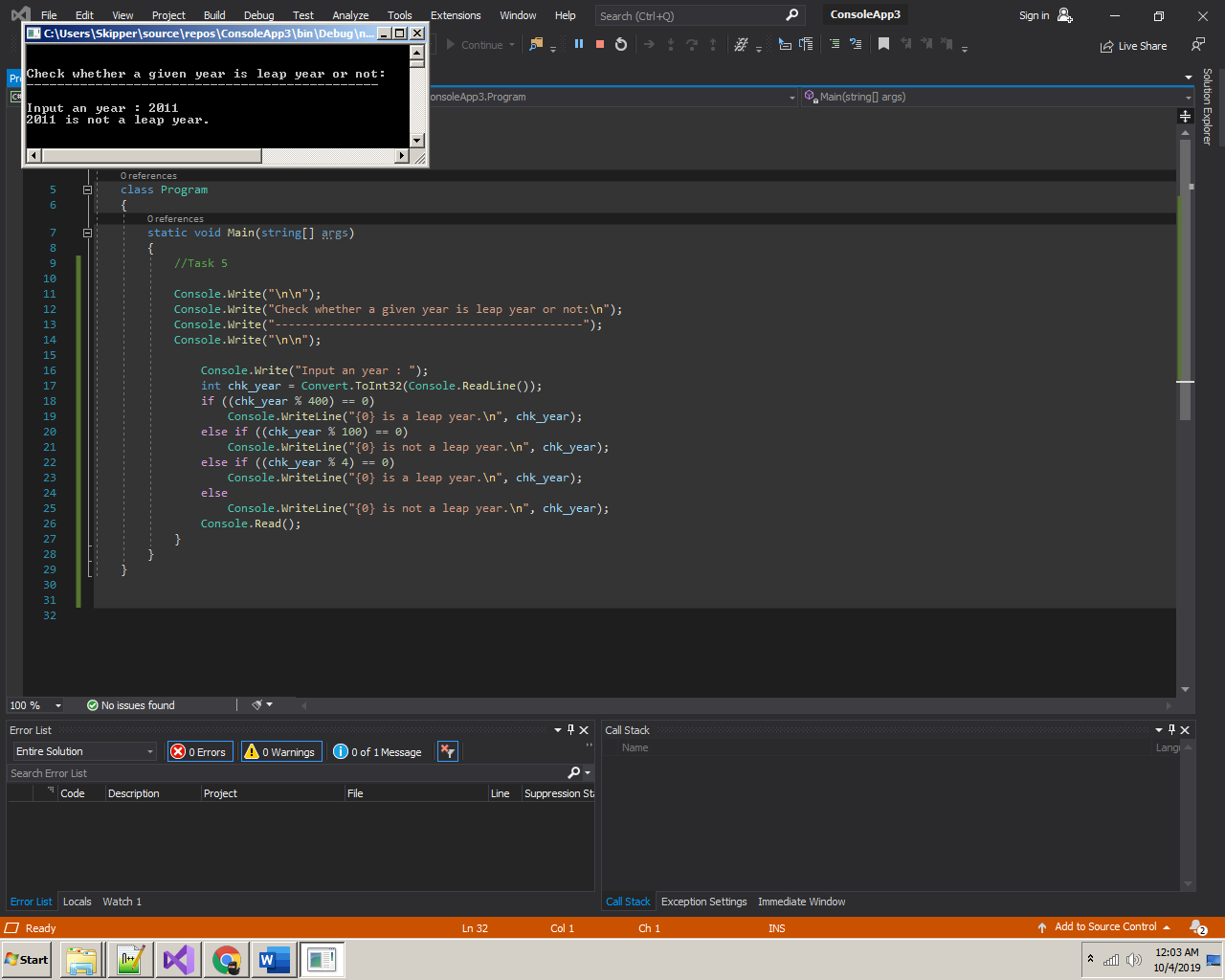
Console.Read();

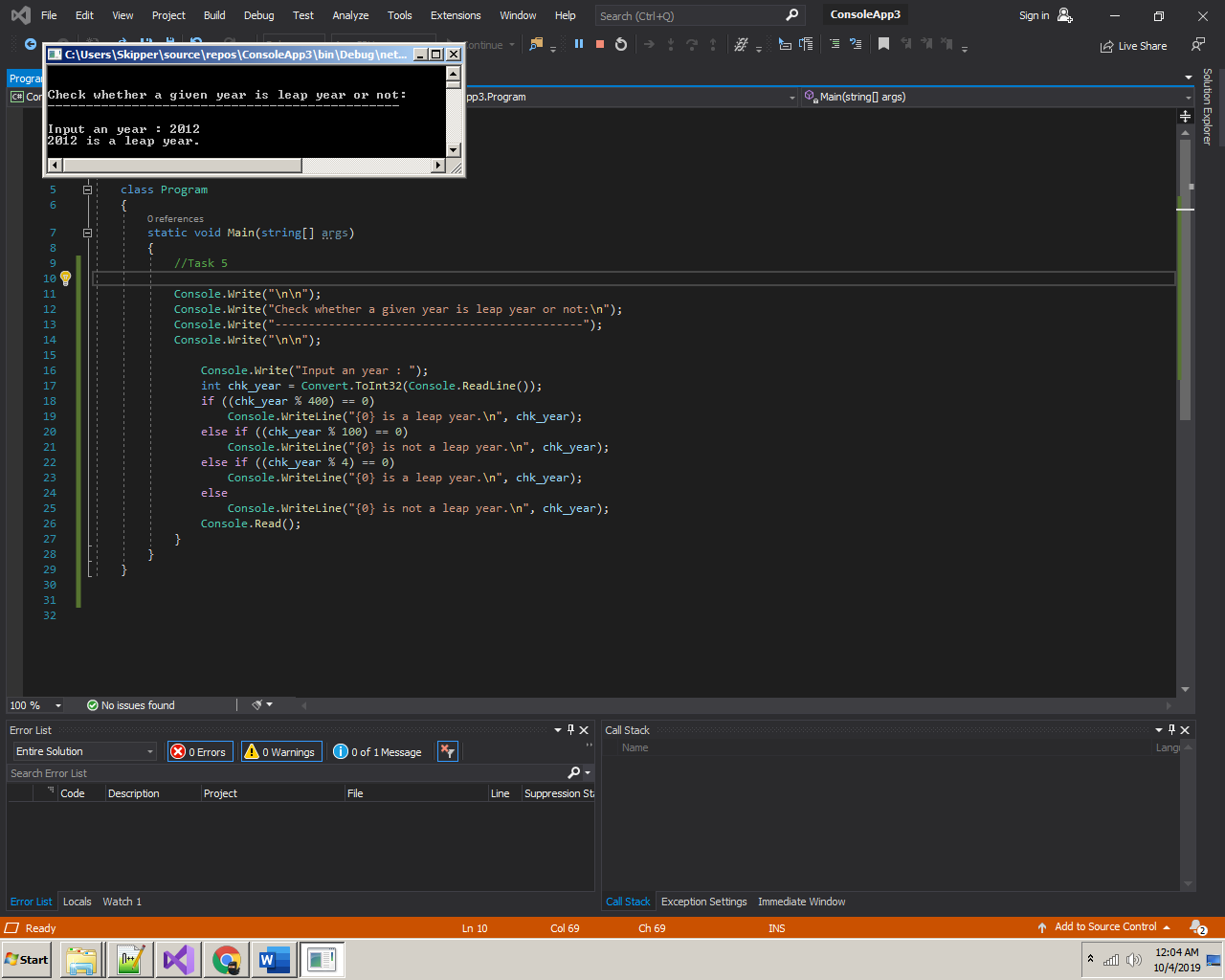
}

}

}

OUTPUT





1. Write a program that takes two numbers as input and an operator as input. Using the switch statement, the program should calculate the result when the operator is applied on the two input numbers.

TASK 6 Solution (CODE)

using System;

namespace VPL\_LAB\_2

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Calculator");

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

Console.WriteLine("+ \n");

Console.WriteLine("- \n");

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

Console.WriteLine("/ \n");

Console.Write("Enter : +,-,\*,/ : \n");

String ch = Console.ReadLine();

int a, b, c;

switch (ch)

{

case "+":

Console.Write("Enter A : ");

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

Console.Write("Enter B : ");

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

c = a + b;

Console.WriteLine("Sum = {0}", c);

break;

case "-":

Console.Write("Enter A : ");

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

Console.Write("Enter B : ");

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

c = a - b;

Console.WriteLine("Difference = {0}", c);

break;

case "\*":

Console.Write("Enter A : ");

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

Console.Write("Enter B : ");

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

c = a \* b;

Console.WriteLine("Product = {0}", c);

break;

case "/":

Console.Write("Enter A : ");

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

Console.Write("Enter B : ");

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

c = a / b;

Console.WriteLine("Quotient = {0}", c);

break;

default:

Console.WriteLine("Invalid Choice");

break;

}

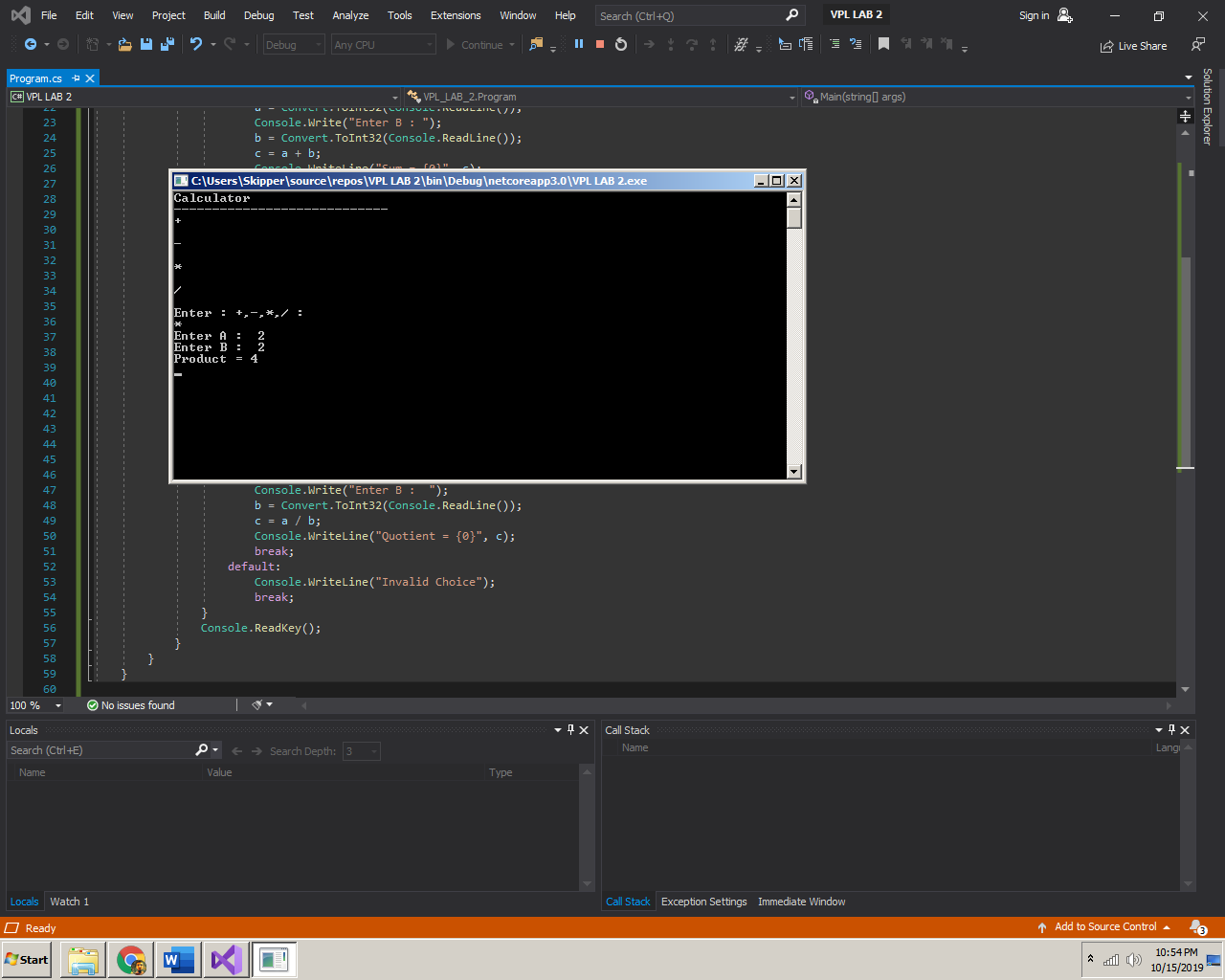
Console.ReadKey();

}

}

}

OUTPUT:



1. Write a program to print Iqra University marks sheet using if…else statement

TASK 2 Solution (CODE)

using System;

namespace VPL\_LAB\_2

{

class Program

{

static void Main(string[] args)

{

int r, m1, m2, m3,m4,m5, t;

float p;

string n;

Console.WriteLine("Enter Roll Number :");

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

Console.WriteLine("Enter Student Name :");

n = Console.ReadLine();

Console.WriteLine("Mark of Subject 1 : ");

m1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Mark of Subject 2 : ");

m2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Mark of Subject 3 : ");

m3 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Mark of Subject 4 : ");

m4 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Mark of Subject 5 : ");

m5 = Convert.ToInt32(Console.ReadLine());

t = m1 + m2 + m3 + m4 + m5;

p = t / 5.0f;

Console.WriteLine("Total : " + t);

Console.WriteLine("Percentage : " + p);

if (p >= 60 && p < 65)

{

Console.WriteLine("Grade is C");

}

else if (p >= 66 && p <= 72)

{

Console.WriteLine("Grade is C+");

}

else if (p > 72 && p <= 78)

{

Console.WriteLine("Grade is B");

}

else if (p > 79 && p <= 86)

{

Console.WriteLine("Grade is B+");

}

else if (p > 87 && p <= 93)

{

Console.WriteLine("Grade is A");

}

else if (p > 93 && p <= 100)

{

Console.WriteLine("Grade is A+");

}

else

{

Console.WriteLine("Grade is F");

}

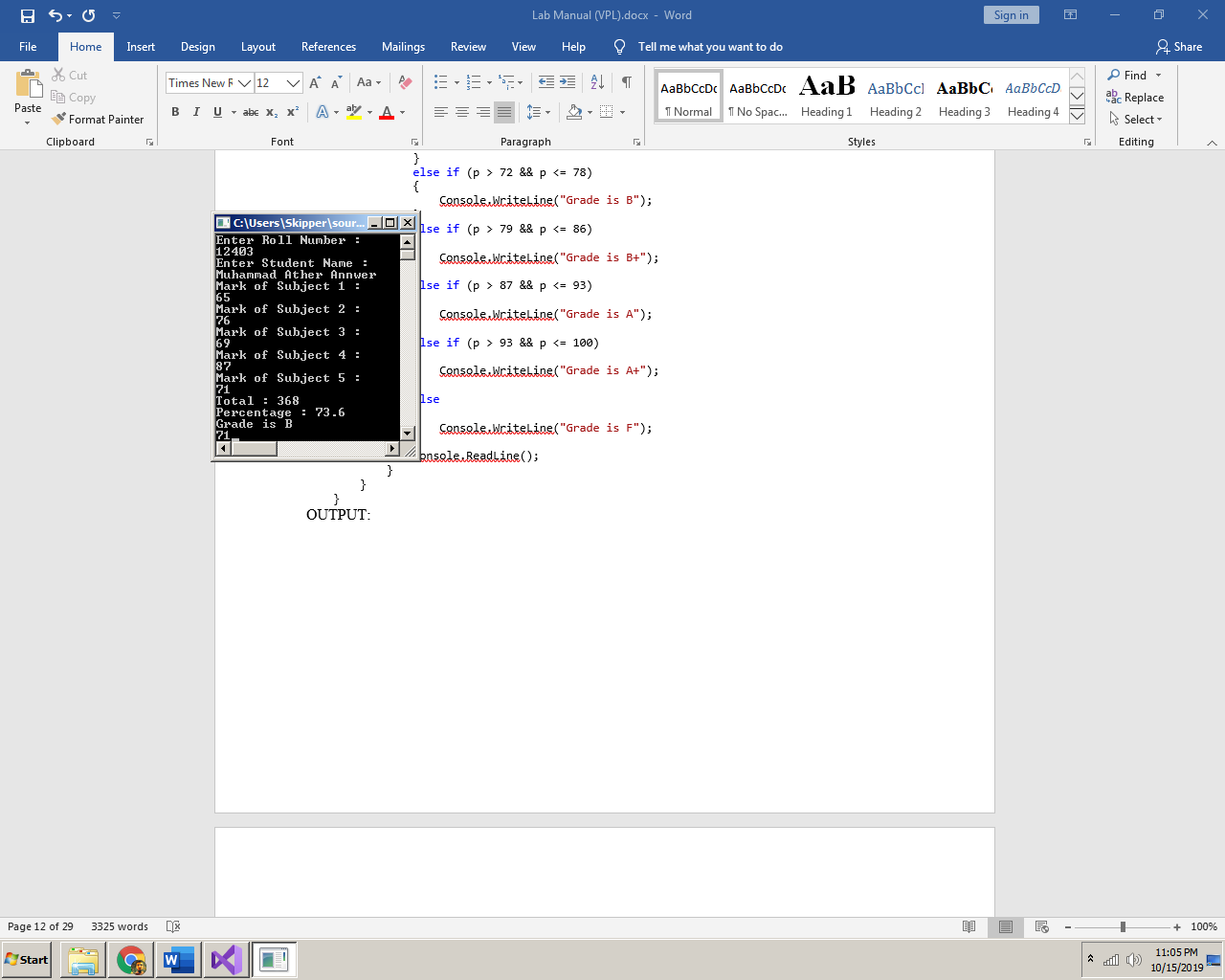
Console.ReadLine();

}

}

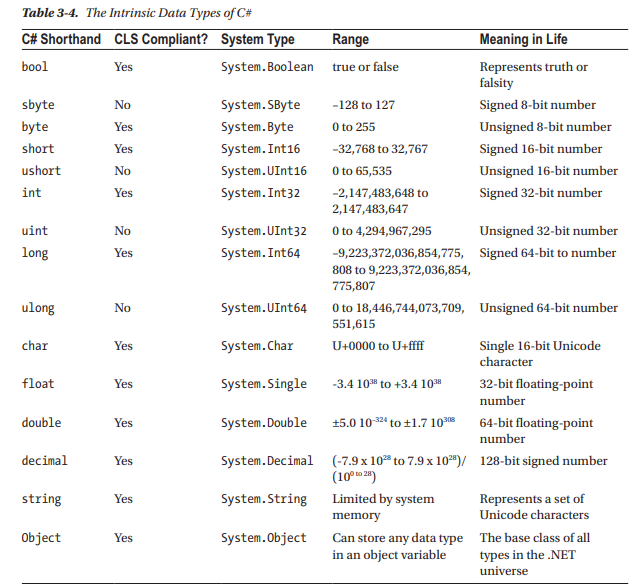
}

OUTPUT:

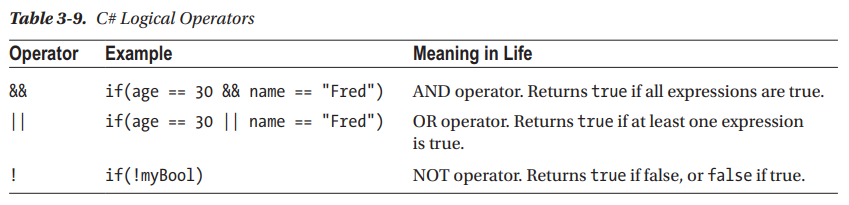


**Lab 2: To study and understand how to write programs in C# using loops, arrays and other constructs**

The objective of this lab is to start writing programs in C# using its basic constructs such as loops, conditions, arrays etc. Following are intrinsic data types supported by C#.



Following are the logical operators in C#:



Loops are used in situations when we need to execute a block of code several number of times. C# has four types of loops: for, foreach, while and do while. An array is a collection of homogeneous data elements. You can declare an array of int as follows:

int[] myInts = new int[3];

**Lab Tasks:**

1. Write a program to count the frequency of each element of an array.

TASK 1 Solution (CODE)

using System;

namespace VPL\_LAB\_2

{

class Program

{

static void Main(string[] args)

{

int[] arr1 = new int[100];

int[] fr1 = new int[100];

int n, i, j, ctr;

Console.Write("\n\nCount the frequency of each element of an array:\n");

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

Console.Write("Input the number of elements to be stored in the array : ");

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

Console.Write("Input {0} elements in the array :\n", n);

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

{

Console.Write("element - {0} : ", i);

arr1[i] = Convert.ToInt32(Console.ReadLine());

fr1[i] = -1;

}

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

{

ctr = 1;

for (j = i + 1; j < n; j++)

{

if (arr1[i] == arr1[j])

{

ctr++;

fr1[j] = 0;

}

}

if (fr1[i] != 0)

{

fr1[i] = ctr;

}

}

Console.Write("\nThe frequency of all elements of the array : \n");

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

{

if (fr1[i] != 0)

{

Console.Write("{0} occurs {1} times\n", arr1[i], fr1[i]);

}

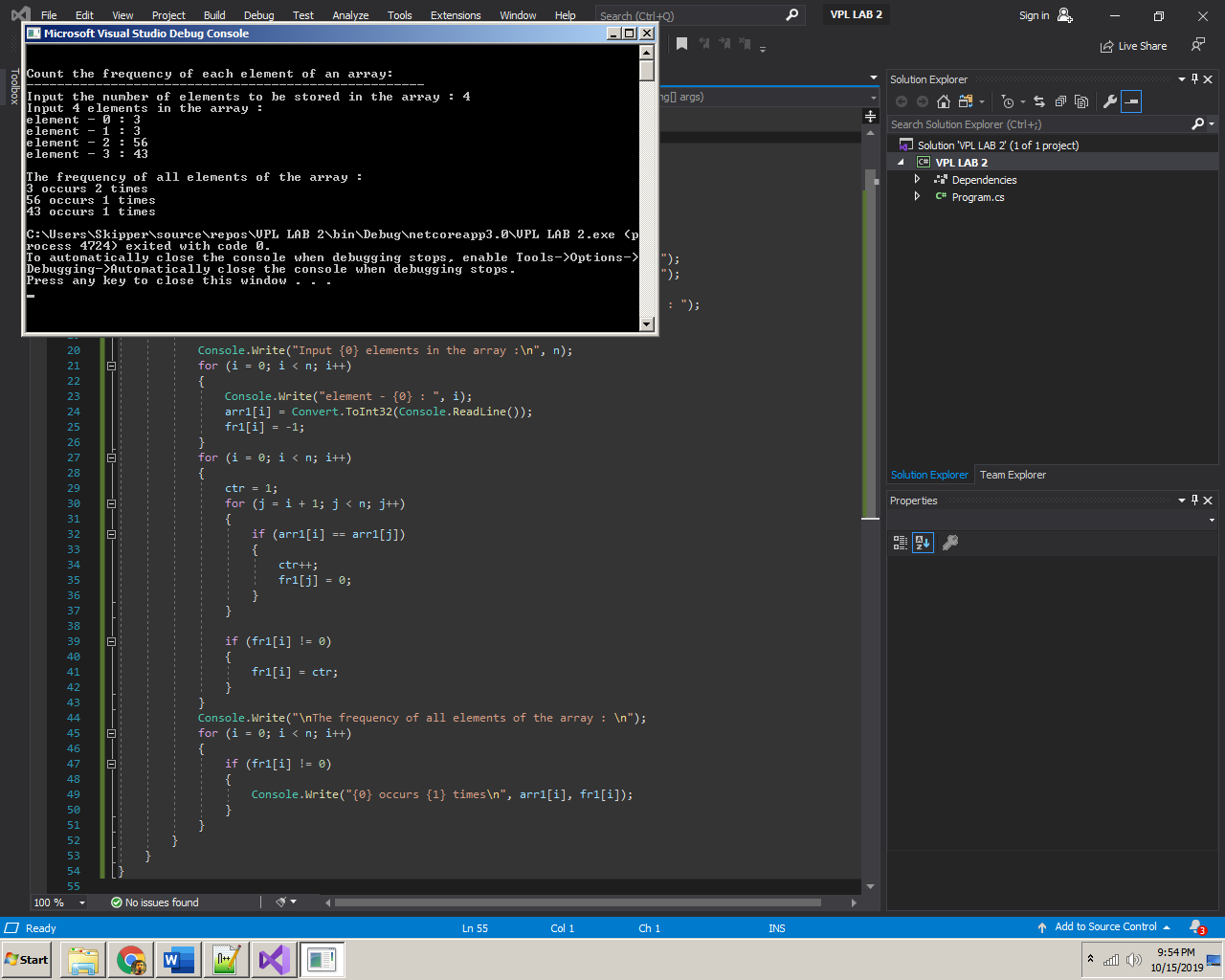
}

}

}

}

OUTPUT:



1. Write a program to find maximum and minimum element in an array.

TASK 2 Solution (CODE)

using System;

namespace VPL\_LAB\_2

{

class Program

{

static void Main(string[] args)

{

int[] arr1 = new int[100];

int i, max, min, n;

Console.Write("\n\nFind maximum and minimum element in an array :\n");

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

Console.Write("Input the number of elements to be stored in the array : ");

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

Console.Write("Input {0} elements in the array : \n", n);

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

{

Console.Write("element - {0} : ", i);

arr1[i] = Convert.ToInt32(Console.ReadLine());

}

max = arr1[0];

min = arr1[0];

for (i = 1; i < n; i++)

{

if (arr1[i] > max)

{

max = arr1[i];

}

if (arr1[i] < min)

{

min = arr1[i];

}

}

Console.Write("Maximum element is : {0}\n", max);

Console.Write("Minimum element is : {0}\n\n", min);

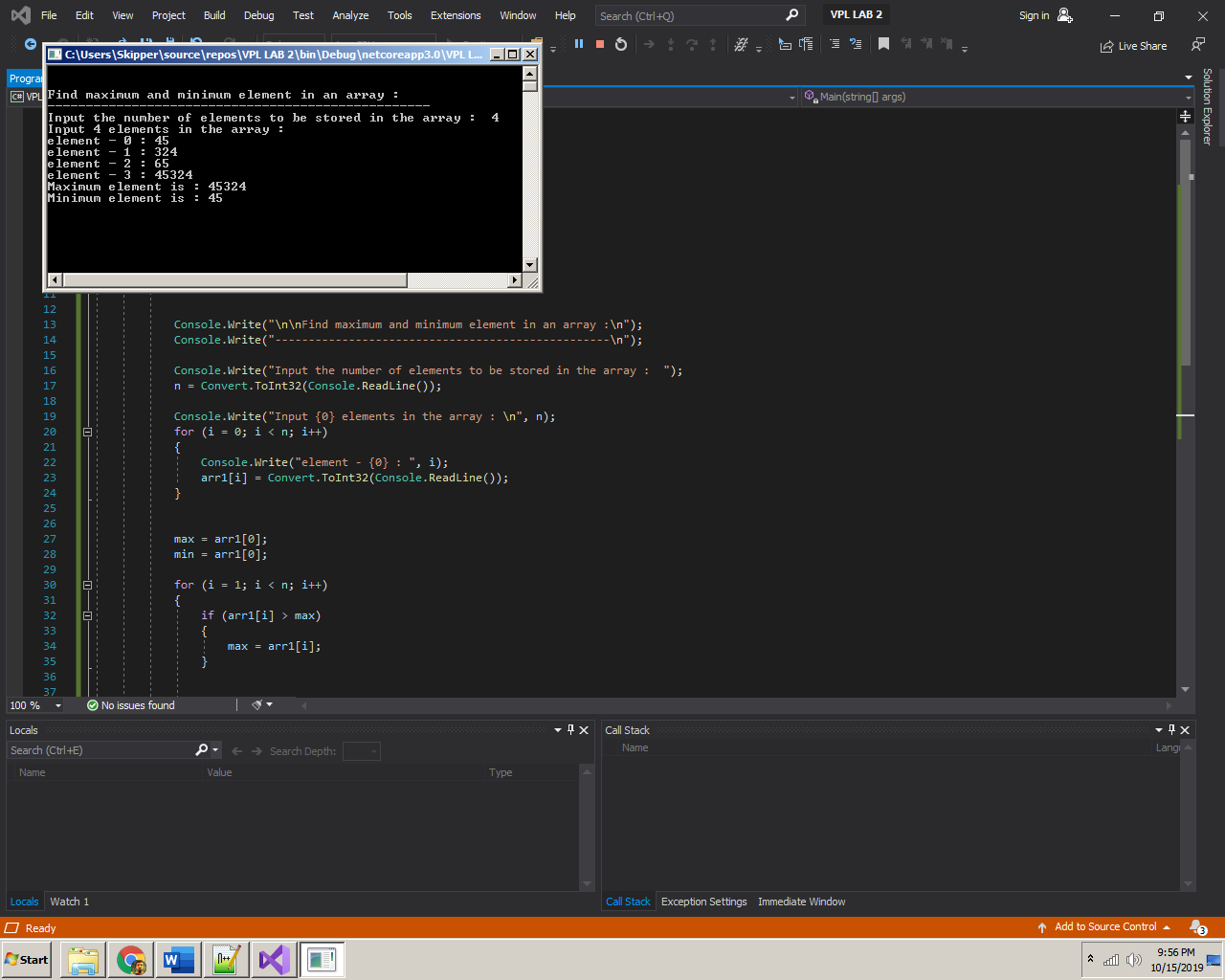
Console.Read();

}

}

}

OUTPUT:



1. Write a program to separate odd and even integers in separate array

TASK 3 Solution (CODE)

using System;

namespace VPL\_LAB\_2

{

class Program

{

static void Main(string[] args)

{

int[] arr1 = new int[10];

int[] arr2 = new int[10];

int[] arr3 = new int[10];

int i, j = 0, k = 0, n;

Console.Write("\n\nSeparate odd and even integers in separate arrays:\n");

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

Console.Write("Input the number of elements to be stored in the array :");

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

Console.Write("Input {0} elements in the array :\n", n);

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

{

Console.Write("element - {0} : ", i);

arr1[i] = Convert.ToInt32(Console.ReadLine());

}

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

{

if (arr1[i] % 2 == 0)

{

arr2[j] = arr1[i];

j++;

}

else

{

arr3[k] = arr1[i];

k++;

}

}

Console.Write("\nThe Even elements are : \n");

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

{

Console.Write("{0} ", arr2[i]);

}

Console.Write("\nThe Odd elements are :\n");

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

{

Console.Write("{0} ", arr3[i]);

}

Console.Write("\n\n");

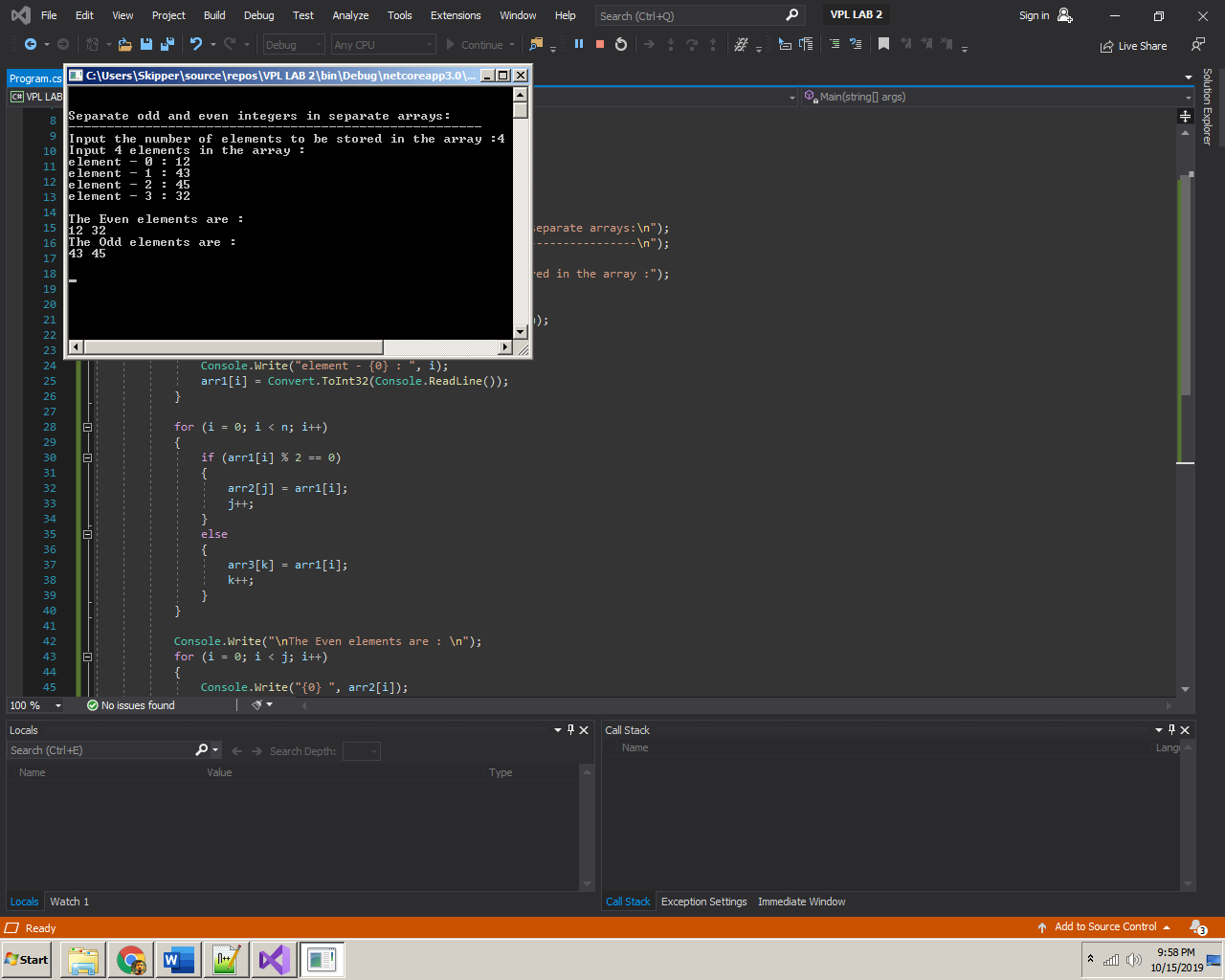
Console.Read();

}

}

}

OUTPUT:



1. Write a program to find the length of a string without using library function.

TASK 4 Solution (CODE)

using System;

namespace VPL\_LAB\_2

{

class Program

{

static void Main(string[] args)

{

string str;

int length = 0;

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

str = Console.ReadLine();

foreach (char chr in str)

{

length += 1;

}

Console.Write("Length of the string is : {0}\n\n", length);

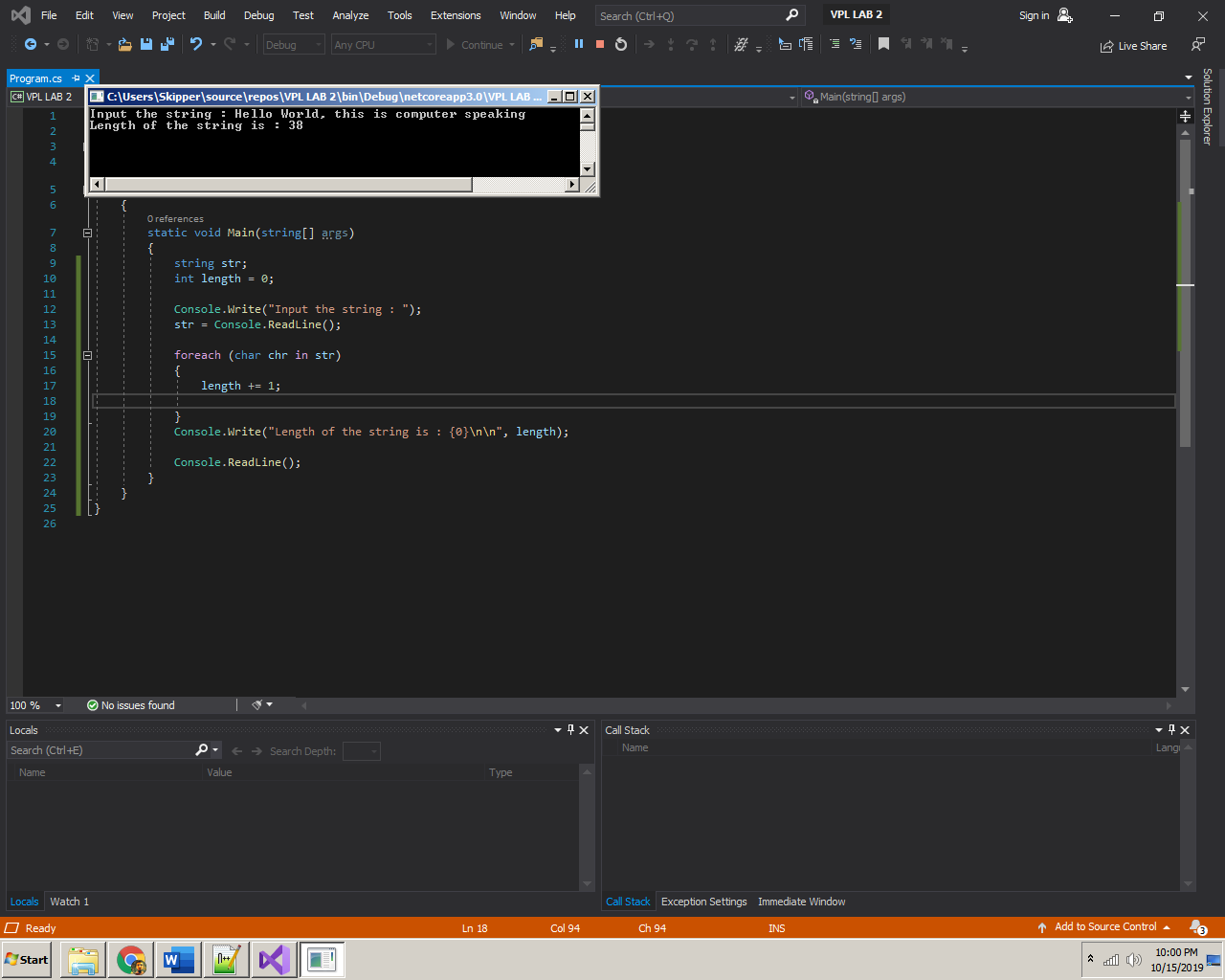
Console.ReadLine();

}

}

}

OUTPUT:



1. Write a program to count the total number of words in a string.

TASK 5 Solution (CODE)

using System;

namespace VPL\_LAB\_2

{

class Program

{

static void Main(string[] args)

{

string str;

int i, wrd, l;

Console.Write("\n\nCount the total number of words in a string :\n");

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

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

str = Console.ReadLine();

l = 0;

wrd = 1;

/\* loop till end of string \*/

while (l <= str.Length - 1)

{

/\* check whether the current character is white space or new line or tab character\*/

if (str[l] == ' ' || str[l] == '\n' || str[l] == '\t')

{

wrd++;

}

l++;

}

Console.Write("Total number of words in the string is : {0}\n", wrd);

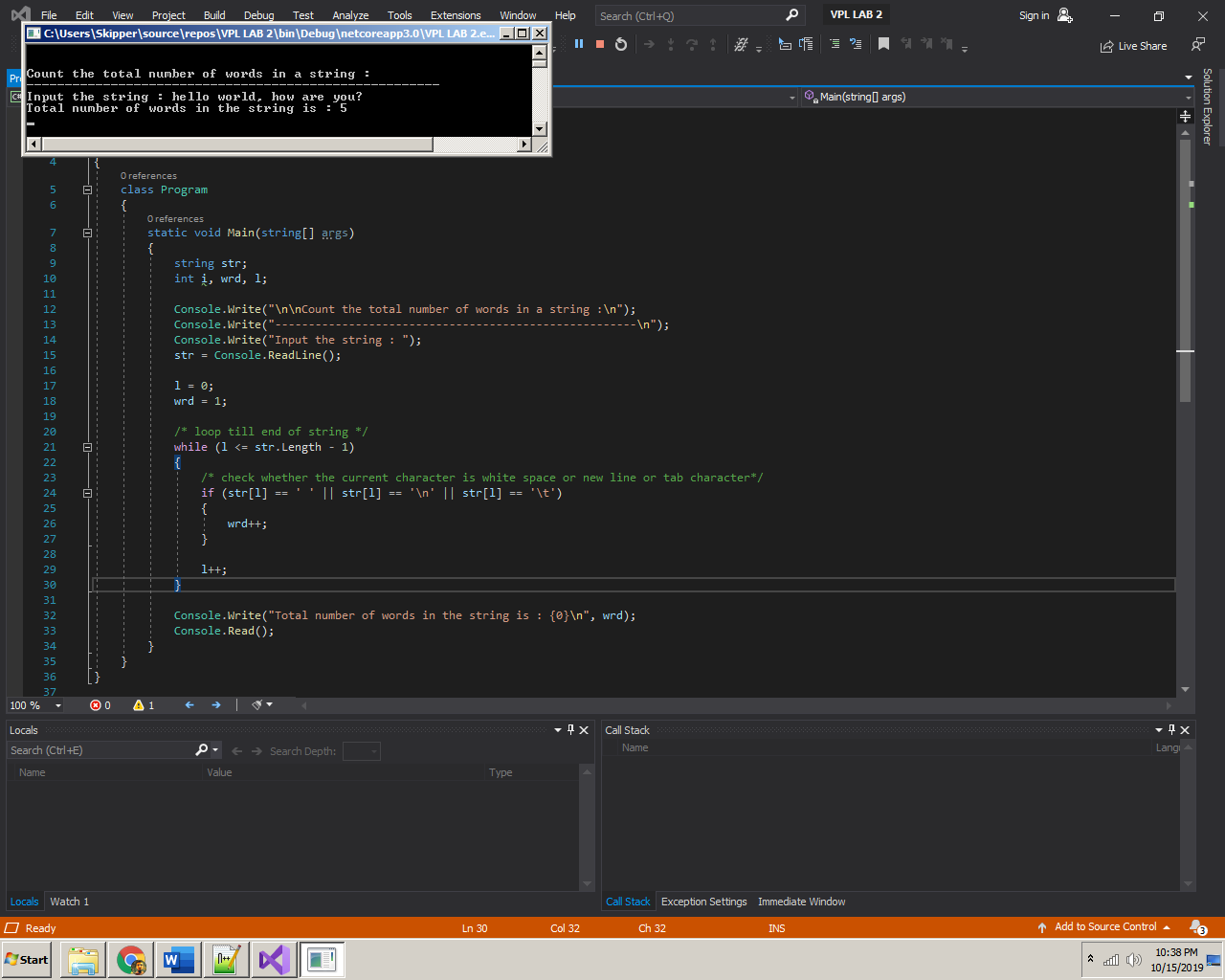
Console.Read();

}

}

}

OUTPUT:



1. Write a program to create a recursive function to calculate the Fibonacci number of a specific term.

TASK 6 Solution (CODE)

using System;

namespace VPL\_LAB\_2

{

class Program

{

public static int Fib(int n1)

{

//if ( (n1 == 1) || (number == 2) )

if (n1 <= 2)

return 1;

else

return Fib(n1 - 1) + Fib(n1 - 2);

}

public static void Main()

{

int num;

Console.Write("\n\nRecursive Function : To calculate the Fibonacci number of a specific term :\n");

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

Console.Write("Enter a number: ");

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

Console.WriteLine("\nThe Fibonacci of {0} th term is {1} \n", num, Fib(num));

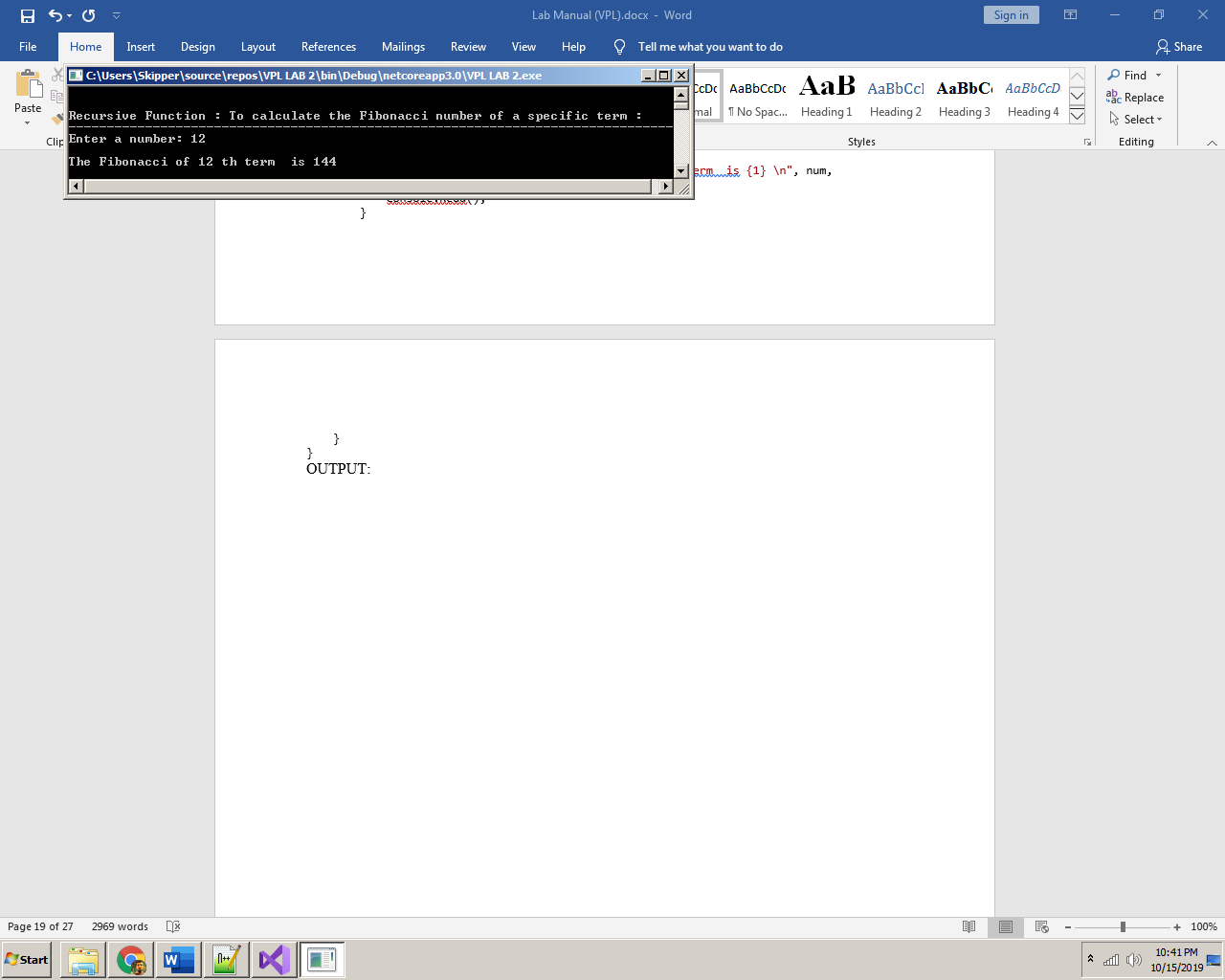
Console.Read();

}

}

}

OUTPUT:



**Lab 3: To study and implement object oriented programming concepts in C#**

Object-oriented programming (OOP) refers to a type of computer programming (software design) in which programmers define not only the data type of a data structure, but also the types of operations (functions) that can be applied to the data structure. In this lab, we are going to implement the three basic pillars of OOP i.e. encapsulation, inheritance and polymorphism.

**Tasks:**

1. Encapsulation
   * + 1. Create a class Circle.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3

{

class Circle

{

private float r;

public float radius

{

get

{

Console.WriteLine("Get: ");

return r;

}

set

{

Console.WriteLine("Set: ");

this.r = value;

}

}

public void setRadius(float value)

{

this.r = value;

}

}

}

* + - 1. Define a property named radius. In the set method, check if the radius is negative then throw an exception System.ArgumentException.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3b

{

class Circle

{

private float r;

public float radius

{

get {

Console.WriteLine("Get: ");

return r;

}

set {

if (this.r >0)

{

this.r = value;

}

else

{

throw new ArgumentException("Negative not allowed");

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3

{

class Program

{

static void Main(string[] args)

{

// Circle obj = new Circle();

//obj.radius = -1;

Student obj = new Student();

obj.id = "12345";

obj.name = "URT";

obj.cgpa = 3.6f;

Console.WriteLine(obj.id);

Console.WriteLine(obj.name);

Console.WriteLine(obj.cgpa);

}

}

}

* + - 1. Define a one argument constructor to initialize the radius.
      2. Define a no argument constructor to initialize the radius value to zero using constructor chaining.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3c

{

class Circle

{

private Double radius;

public Double Radius

{

get {

return radius;

}

set {

if (value > 0){

radius = value;

}else{

throw new System.ArgumentException("Parameter cannot be < than 0", "original");;

}

}

}

public Circle(Double radius)

{

}

public Circle()

{

this.radius = 0.0;

}

public Double getArea(Double rad) {

return 3.14 \* (rad \* rad);

}

}

class Program

{

static void Main(string[] args)

{

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

Circle a = new Circle();

try {

a.Radius = val;

Console.WriteLine(a.getArea(a.Radius));

}

catch(ArgumentException ex){

Console.WriteLine(ex);

}

Console.ReadLine();

}

}

}

* + - 1. Define a method GetArea() to calculate the area of circle.

class Program

{

static void Main(string[] args)

{

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

Circle a = new Circle();

try {

a.Radius = val;

Console.WriteLine(a.getArea(a.Radius));

}

catch(ArgumentException ex){

Console.WriteLine(ex);

}

Console.ReadLine();

}

}

}

* + - 1. Create a Test class.
      2. In the Main method, define two objects of Circle and initialize them with random values.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3d

{

class Circle

{

private Double radius;

public Double Radius

{

get {

return radius;

}

set {

if (value > 0){

radius = value;

}else{

throw new System.ArgumentException("Parameter cannot be < 0", "original");;

}

}

}

public Circle(Double radius)

{

}

public Circle()

{

this.radius = 0.0;

}

public Double getArea(Double rad) {

return 3.14 \* (rad \* rad);

}

}

class Program

{

static void Main(string[] args)

{

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

Circle a = new Circle();

try {

a.Radius = val;

Console.WriteLine(a.getArea(a.Radius));

}

catch(ArgumentException ex){

Console.WriteLine(ex);

}

Console.ReadLine();

}

}

}

* + - 1. Call the GetArea() method of each object and print the area.

1. Inheritance/ Polymorphism
   * + 1. Create an abstract class Animal. Define a property: name of type string. Define an abstract method sound(). Define a constructor to initialize the animal’s name.
       2. Now define two abstract classes named Mammal and Non-Mammal that inherits the Animal class.
       3. Inherit the Mammal class to define Cat class. Implement the method sound that prints ‘Meow’.
       4. Implement classes for the Goat inheriting the Mammal and the Fish inheriting the Non-Mammal class
       5. Create a Test class. Define few objects of classes Cat, Goat and Fish. Assign the instance variables to reference variable of Animal class and polymorphically call them.
2. **Code:**
3. using System;
4. using System.Collections.Generic;
5. using System.Linq;
6. using System.Text;
7. using System.Threading.Tasks;
8. namespace ConsoleApplication7
9. {
10. class Program
11. {
12. static void Main(string[] args)
13. {
14. Circle c = new Circle();
15. //c.radius = 10;
16. Console.WriteLine(c.radius);
17. }
18. }
19. }
20. **Code circle class:**
21. using System;
22. using System.Collections.Generic;
23. using System.Linq;
24. using System.Text;
25. using System.Threading.Tasks;
26. namespace ConsoleApplication7
27. {
28. class Circle
29. {
30. private float r;
31. public float radius
32. {
33. get
34. {
35. return r;
36. }
37. set
38. {
39. this.r = value;
40. }
41. }
42. }
43. }

**Lab 4: To study and implement Windows Forms application in C#**

In this lab we will implement windows forms in C#. Windows Forms (WinForms) is a graphical (GUI) class library included as a part of Microsoft .NET Framework or Mono Framework, providing a platform to write rich client applications for desktop, laptop, and tablet PCs.

We will look at some of the most common controls used in C# and use them to develop a basic application.

**Tasks:**

1. Design a basic Sign Up form with fields name, id, password, email, address, gender and date of birth.
2. Add a checkbox to accept license agreement, and a view button to view the license agreement.
3. Add a Sign Up button.
4. When the user clicks on Sign Up, following validations are to be performed:
   1. Password must contain a capital letter and one digit
   2. Email address should be in proper format
   3. All the fields must be filled by user
   4. The user must be 18 years or above
   5. The user has accepted the license agreement
5. When the user clicks on view license agreement button, a new form is to be displayed with the license agreement.

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 WindowsFormsApplication1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

private void btn\_signup\_Click(object sender, EventArgs e)

{

string name = this.txtbox\_username.Text;

string user\_id = this.txtbox\_id.Text;

string email = this.txtbox\_email.Text;

string dob = this.dtpick\_DOB.Text;

string pass = this.txtbox\_pass.Text;

string gender = "Male";

if (this.Rbtn\_female.Checked)

{

gender = "Female";

}

else if (this.Rbtn\_male.Checked)

{

gender = "Male";

}

bool license\_aggreement = this.CHKBOX\_license.Checked;

if (name == "" ||

user\_id == "" ||

email == "" ||

dob == "" ||

pass == "" ||

license\_aggreement == false)

{

MessageBox.Show("Please fill all the fields.");

return;

}

if (!email.Contains("@") ||

!email.Contains("."))

{

MessageBox.Show("Please Provide a valid Email Address.");

return;

}

string patternPassword = @"^(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{4,9}$";

if (!string.IsNullOrEmpty(pass))

{

if (!System.Text.RegularExpressions.Regex.IsMatch(pass, patternPassword))

{

MessageBox.Show("Password must be at least 4 characters, " +

"no more than 9 characters, and must include at least " +

"one upper case letter, one lower case letter," +

" and one numeric digit.");

return;

}

}

}

private void btn\_viewlicense\_Click(object sender, EventArgs e)

{

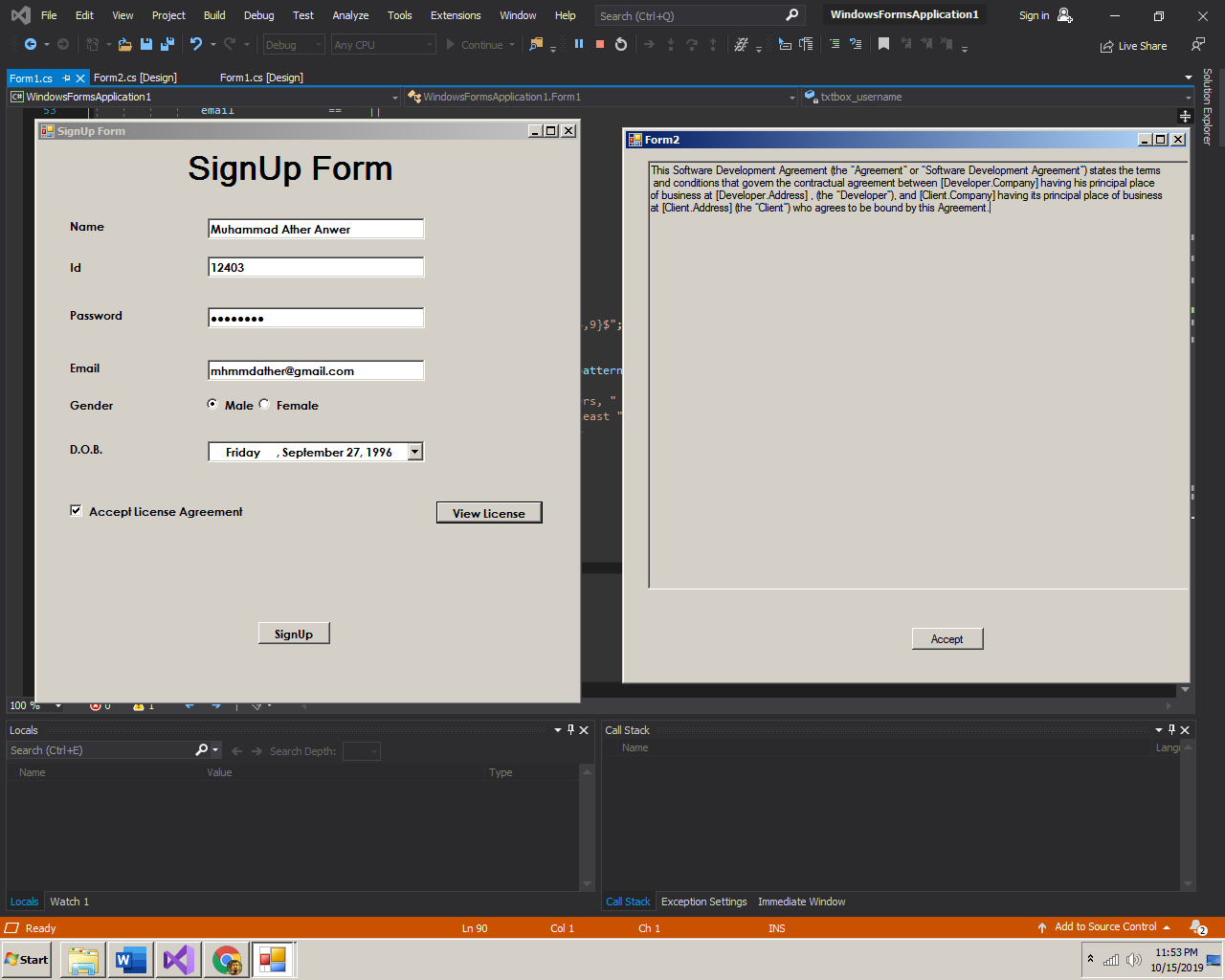
Form2 f = new Form2();

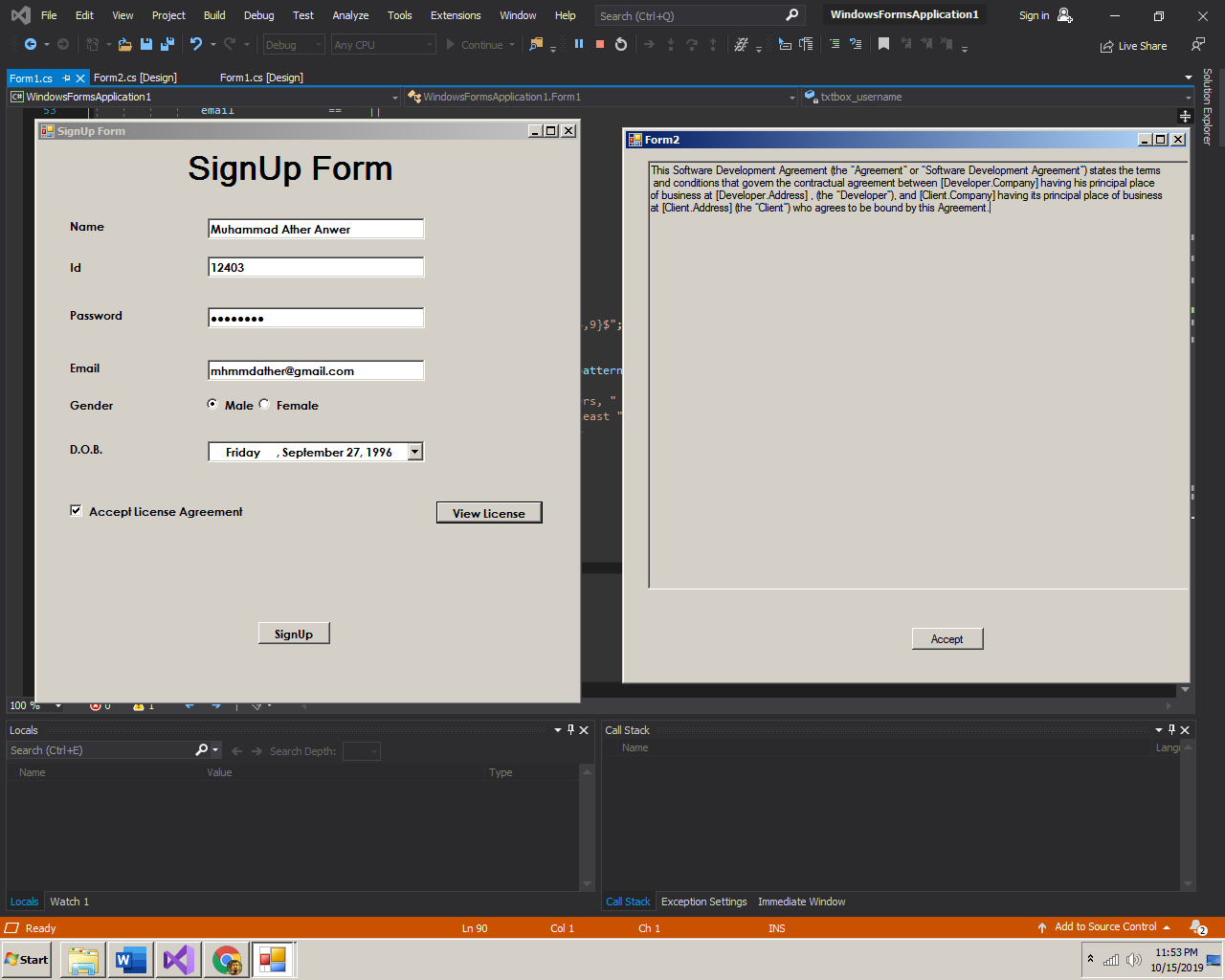
f.Show();

}

}

}





**Lab 5: To study and implement Collections in C#**

In todays, lab we will implement the collections in C#. A collection is an abstract data type for grouping together multiple values. It's therefore sometime known as container. A collection is just a grouping of some objects with the same type.

**Lab Tasks:**

1. Create a WinForm in C# with following buttons: Add a Student, Edit a Student, Delete a Student, Search a Student.
2. Now, create a class Student with the following properties: name, age, gender, cgpa
3. Implement the Add a Student button. When the user clicks this button, open a new form from which user can provide the details. Save the results in a collection.
4. Implement the Edit button such that user can modify the students record. The changes are reflected in the collection.
5. Implement the Delete and Search options.

Solution:

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 WindowsFormsApplication2

{

public partial class AddStudent : Form

{

private List<Student> students;

private Student s;

public AddStudent(List<Student> students)

{

this.students = students;

InitializeComponent();

}

public AddStudent(Student s)

{

this.s = s;

InitializeComponent();

this.loadData();

this.adjustControls();

}

public void adjustControls()

{

this.Text = "Edit Student Data";

this.button2.Text = "Edit";

this.sname.Enabled = false;

}

public void loadData()

{

this.sname.Text = s.name;

this.age.Text = s.age.ToString();

this.cgpa.Text = s.cgpa.ToString();

if (s.gender.Equals("Male"))

{

this.gender.SelectedIndex = 0;

}

else

{

this.gender.SelectedIndex = 1;

}

}

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

{

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

if (button2.Text == "Add")

{

Student s = new Student();

s.name = this.sname.Text;

s.age = int.Parse(this.age.Text);

s.cgpa = float.Parse(this.cgpa.Text);

s.gender = this.gender.SelectedItem.ToString();

students.Add(s);

MessageBox.Show("Student added successfully");

}

else

{

s.name = this.sname.Text;

s.age = int.Parse(this.age.Text);

s.cgpa = float.Parse(this.cgpa.Text);

s.gender = this.gender.SelectedItem.ToString();

MessageBox.Show("Student edited successfully");

}

this.Close();

}

private void AddStudent\_Load(object sender, EventArgs e)

{

}

}

}

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 WindowsFormsApplication2

{

public partial class ListStudent : Form

{

List<Student> students;

public ListStudent(List<Student> students)

{

this.students = students;

InitializeComponent();

}

private void ListStudent\_Load(object sender, EventArgs e)

{

this.student\_list.DataSource = this.students;

}

public void select\_edit()

{

this.Edit.Checked = true;

}

public void select\_delete()

{

this.Delete.Checked = true;

}

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

{

string name = this.student\_list.SelectedItem.ToString();

Student s = new Student();

s.name = name;

int i = students.IndexOf(s);

if (Delete.Checked)

{

students.RemoveAt(i);

this.student\_list.DataSource = null;

this.student\_list.DataSource = students;

}

else

{

s = students[i];

AddStudent add = new AddStudent(s);

add.Show();

}

}

}

}

using System;

using System.Collections;

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 WindowsFormsApplication2

{

public partial class Form1 : Form

{

private List<Student> students = new List<Student>();

public Form1()

{

InitializeComponent();

}

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

{

AddStudent add = new AddStudent(students);

add.Show();

}

private void Form1\_Load(object sender, EventArgs e)

{

Student s1 = new Student();

s1.name = "Ali";

s1.gender = "Male";

s1.age = 21;

s1.cgpa = 3.6f;

students.Add(s1);

s1 = new Student();

s1.name = "Areeba";

s1.gender = "Female";

s1.age = 20;

s1.cgpa = 3.2f;

students.Add(s1);

}

private void button2\_Click(object sender, EventArgs e)

{

ListStudent ls = new ListStudent(this.students);

ls.select\_edit();

ls.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

ListStudent ls = new ListStudent(this.students);

ls.select\_delete();

ls.Show();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WindowsFormsApplication2

{

public class Student

{

public string name { get; set; }

public int age { get; set; }

public float cgpa { get; set; }

public string gender { get; set; }

public override String ToString()

{

return name;

}

public override bool Equals(object obj)

{

if (!Convert.IsDBNull(obj))

{

Student s = (Student)obj;

return this.name.Equals(s.name);

}

return false;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApplication2

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

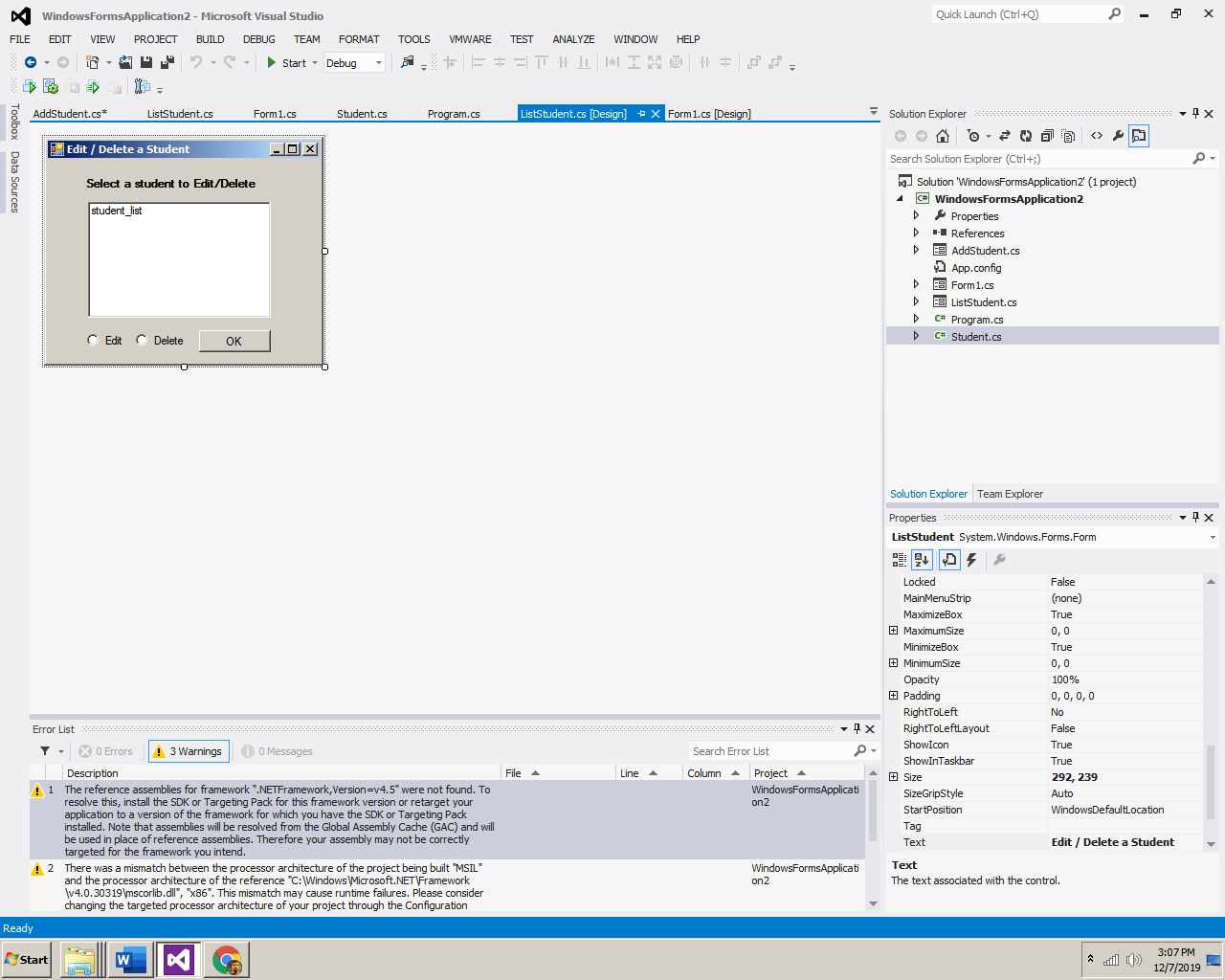
Application.SetCompatibleTextRenderingDefault(false);

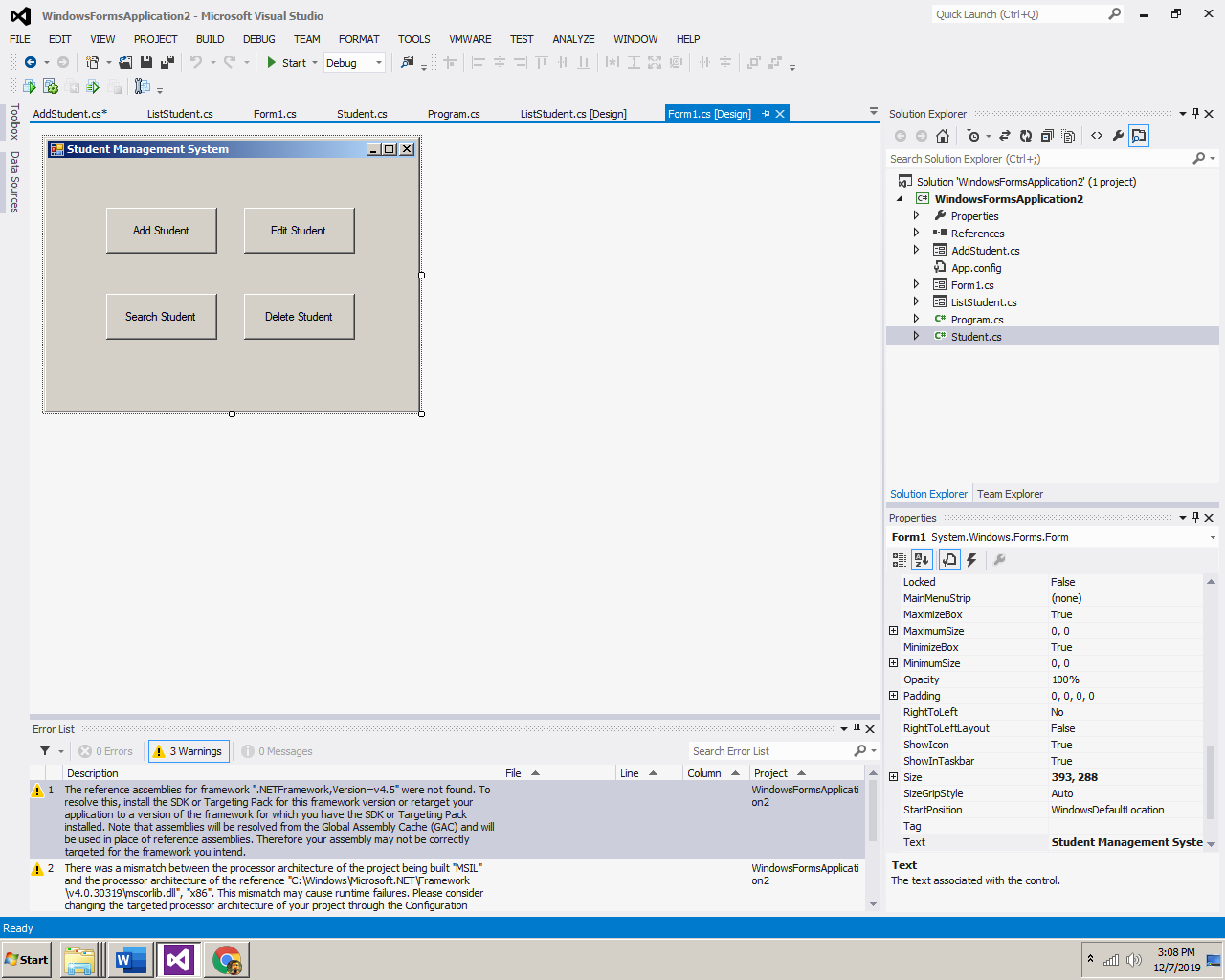
Application.Run(new Form1());

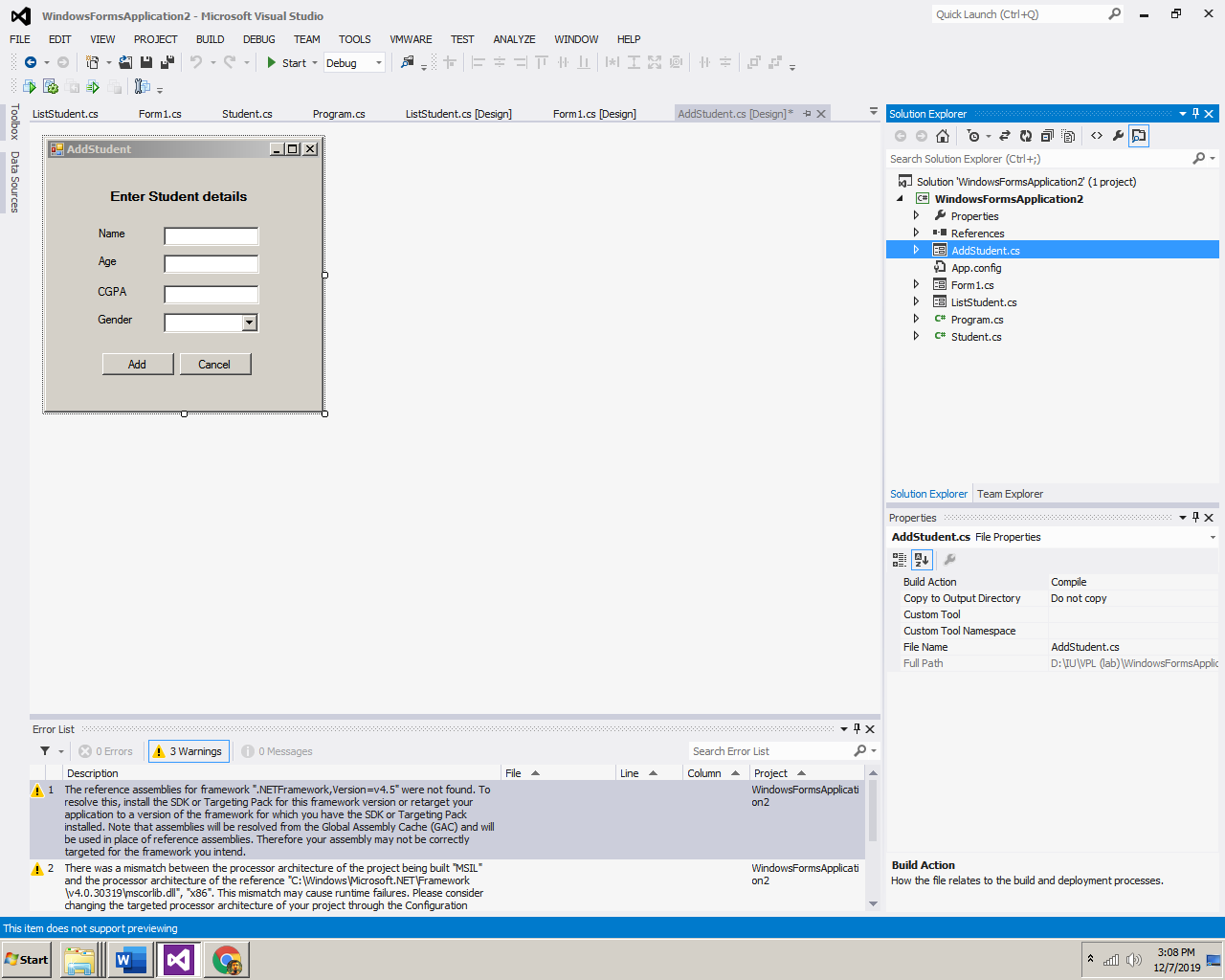
}

}

}







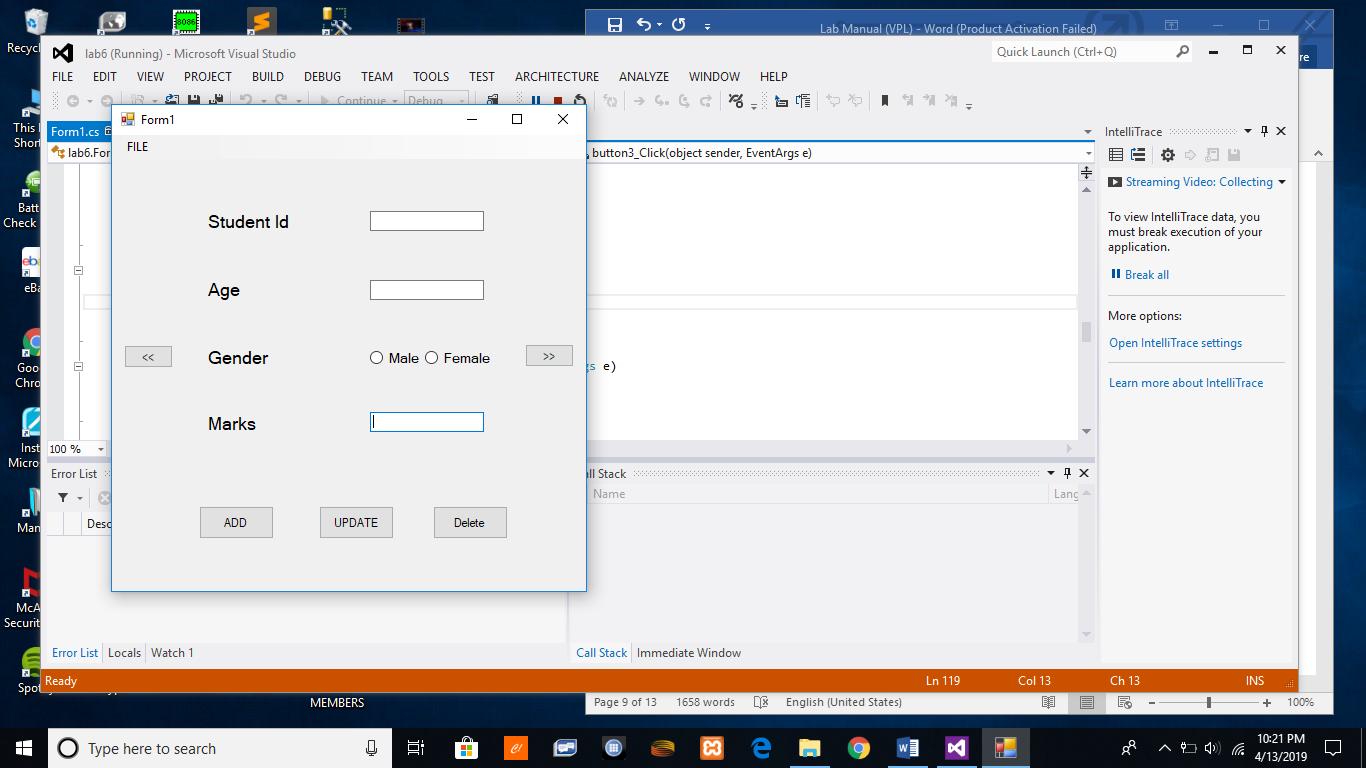
**Lab 6: To study and implement I/O in C#**

In this lab, we will use StreamReader and StreamWriter of C# to develop a basic application that can insert student’s records in a file. A stream can be defined as a sequence of data. The StreamReader is used to read data from a source and the StreamWriter is used for writing data to a destination.

**Lab Tasks:**

1. Create a basic form in C# with the following fields: student id, age, gender, marks.
2. Add the buttons to add a new record, delete a record and update a record, in memory.
3. Create a Menu Strip from which user can save the records in a file or load the records from a file.
4. Code:
5. using System;
6. using System.Collections.Generic;
7. using System.ComponentModel;
8. using System.Data;
9. using System.Drawing;
10. using System.Linq;
11. using System.Text;
12. using System.Threading.Tasks;
13. using System.Windows.Forms;
14. using System.IO;
15. namespace lab6
16. {
17. public partial class Form1 : Form
18. {
19. List<Student> students = new List<Student>();
20. int current =-1;
21. public Form1()
22. {
23. InitializeComponent();
24. }
25. private void Form1\_Load(object sender, EventArgs e)
26. {
27. }
28. private void button1\_Click(object sender, EventArgs e)
29. {
30. Student std = new Student();
31. std.id = textBox1.Text;
32. std.age = textBox2.Text;
33. if (Male.Checked)
34. {
35. std.gender = "male";
36. }
37. else {
38. std.gender = "female";
39. }
40. std.marks = textBox3.Text;
41. students.Add(std);
42. current = students.Count - 1;
43. MessageBox.Show("Student added successfully");
45. }
46. private void button5\_Click(object sender, EventArgs e)
47. {
49. if (current >= students.Count-1)
50. {
51. return;
53. }
54. else {
55. current++;
56. Student s = students[current];
57. textBox1.Text = s.id;
58. textBox2.Text = s.age;
59. textBox3.Text = s.marks;
60. if (s.gender == "male")
61. {
62. Male.Checked = true;
63. }
64. else
65. {
66. Female.Checked = true; } }
67. }
68. private void button4\_Click(object sender, EventArgs e)
69. {
70. if (current <= 0)
71. {
72. return;
73. }
74. else {
75. current--;
76. Student s = students[current];
77. textBox1.Text = s.id;
78. textBox2.Text = s.age;
79. textBox3.Text = s.marks;
80. if (s.gender == "male")
81. {
82. Male.Checked = true;
83. }
84. else
85. {
86. Female.Checked = true;
87. }
88. }
89. }
90. private void button2\_Click(object sender, EventArgs e)
91. {
92. Student s = students[current];
93. s.id = textBox1.Text;
94. s.age = textBox2.Text;
95. s.marks = textBox3.Text;
96. if (Male.Checked)
97. {
98. s.gender = "Male";
99. }
100. else {
101. s.gender = "Female";
102. }
103. }
104. private void button3\_Click(object sender, EventArgs e)
105. {
106. students.RemoveAt(current);
107. current--;
108. }
109. private void oPENToolStripMenuItem\_Click(object sender, EventArgs e)
110. {
111. }
112. private void eXITToolStripMenuItem1\_Click(object sender, EventArgs e)
113. {
114. this.Close();
115. }
116. private void eXITToolStripMenuItem\_Click(object sender, EventArgs e)
117. {
118. if (saveFileDialog1.ShowDialog() == DialogResult.OK)
119. {
120. StreamWriter sw = new StreamWriter(saveFileDialog1.FileName,true);
121. foreach (Student s in students)
122. {
123. sw.WriteLine(s.id + " " + s.age + " " + s.marks+ " "+s.gender);
124. }
125. sw.Close();
126. MessageBox.Show("File Saved Successfully");
127. }
128. }
129. private void saveFileDialog1\_FileOk(object sender, CancelEventArgs e)
130. {
131. }
132. private void sAVEToolStripMenuItem\_Click(object sender, EventArgs e)
133. {
134. if (openFileDialog1.ShowDialog() == DialogResult.OK)
135. {
136. StreamReader sr = new StreamReader(openFileDialog1.FileName,true);
138. string line = sr.ReadLine();
139. students = new List<Student>();
141. while (line != null)
142. {
143. string[] tokens = line.Split();
144. Student s = new Student();
145. s.id = tokens[0];
146. s.age = tokens[1];
147. s.marks = tokens[2];
148. s.gender = tokens[3];
149. students.Add(s);
150. line = sr.ReadLine();
151. }
152. current = 0; } } }}
153. STUDENT CLASS:
154. using System;
155. using System.Collections.Generic;
156. using System.Linq;
157. using System.Text;
158. using System.Threading.Tasks;
159. namespace lab6
160. {
161. class Student
162. {
163. public string id;
164. public string age;
165. public string gender;
166. public string marks;
167. }
168. }

Output



**Lab 7: To study and implement XML parsing in C#**

Extensible Markup Language (XML) defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is a software- and hardware-independent tool for storing and transporting data

**Lab Tasks:**

1. Create a form with a text field centered on window.
2. Create a menu strip to load XML configuration file.
3. Create an XML file to store the following information:

* Window size: The size of the current window
* Window title: The title of the window
* Background color: The background color of the window
* Foreground color: The foreground color of the window

1. You should be able to apply the settings stored in XML file to the window.

Code:

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.Xml;

namespace WindowsFormsApplication4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

this.Close();

}

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

{

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

string fname = openFileDialog1.FileName;

XmlDocument d = new XmlDocument();

d.Load(fname);

string text = d.GetElementsByTagName("text")[0].InnerText;

this.textBox1.Text = text;

string title = d.GetElementsByTagName("title")[0].InnerText;

this.Text = text;

string bcolor = d.GetElementsByTagName("background-color")[0].InnerText;

this.BackColor = Color.FromName(bcolor);

string fcolor = d.GetElementsByTagName("foreground-color")[0].InnerText;

this.label1.ForeColor = Color.FromName(fcolor);

}

}

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

{

this.colorDialog1.ShowDialog();

this.BackColor = this.colorDialog1.Color;

this.button1.ForeColor = this.colorDialog1.Color;

}

private void button2\_Click(object sender, EventArgs e)

{

this.colorDialog1.ShowDialog();

this.label1.ForeColor= this.colorDialog1.Color;

this.button2.ForeColor = this.colorDialog1.Color;

}

private void button3\_Click(object sender, EventArgs e)

{

this.Text = textBox2.Text;

}

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

{

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

XmlDocument d = new XmlDocument();

XmlText text = d.CreateTextNode(this.textBox1.Text);

XmlText title = d.CreateTextNode(this.Text);

XmlText fg = d.CreateTextNode(this.BackColor.ToArgb().ToString());

XmlText bg = d.CreateTextNode(this.ForeColor.ToArgb().ToString());

XmlElement conf = d.CreateElement(string.Empty, "configuration", string.Empty);

d.AppendChild(conf);

XmlElement el = d.CreateElement(string.Empty, "text", string.Empty);

el.AppendChild(text);

conf.AppendChild(el);

el = d.CreateElement(string.Empty, "title", string.Empty);

el.AppendChild(title);

conf.AppendChild(el);

el = d.CreateElement(string.Empty, "background-color", string.Empty);

el.AppendChild(bg);

conf.AppendChild(el);

el = d.CreateElement(string.Empty, "foreground-color", string.Empty);

el.AppendChild(fg);

conf.AppendChild(el);

d.Save(saveFileDialog1.FileName);

}

}

private void Form1\_Load(object sender, EventArgs e)

{

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApplication4

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

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.Xml;

namespace WindowsFormsApplication2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

XmlDocument d = new XmlDocument();

d.Load(openFileDialog1.FileName);

XmlNodeList l = d.GetElementsByTagName("title");

this.Text = l[0].InnerText;

l = d.GetElementsByTagName("Text");

this.textBox1.Text = l[0].InnerText;

l = d.GetElementsByTagName("Width");

this.Width = int.Parse(l[0].InnerText);

l = d.GetElementsByTagName("Height");

this.Height= int.Parse(l[0].InnerText);

l = d.GetElementsByTagName("Background");

this.BackColor= Color.FromName(l[0].InnerText);

l = d.GetElementsByTagName("Foreground");

this.textBox1.ForeColor = Color.FromName(l[0].InnerText);

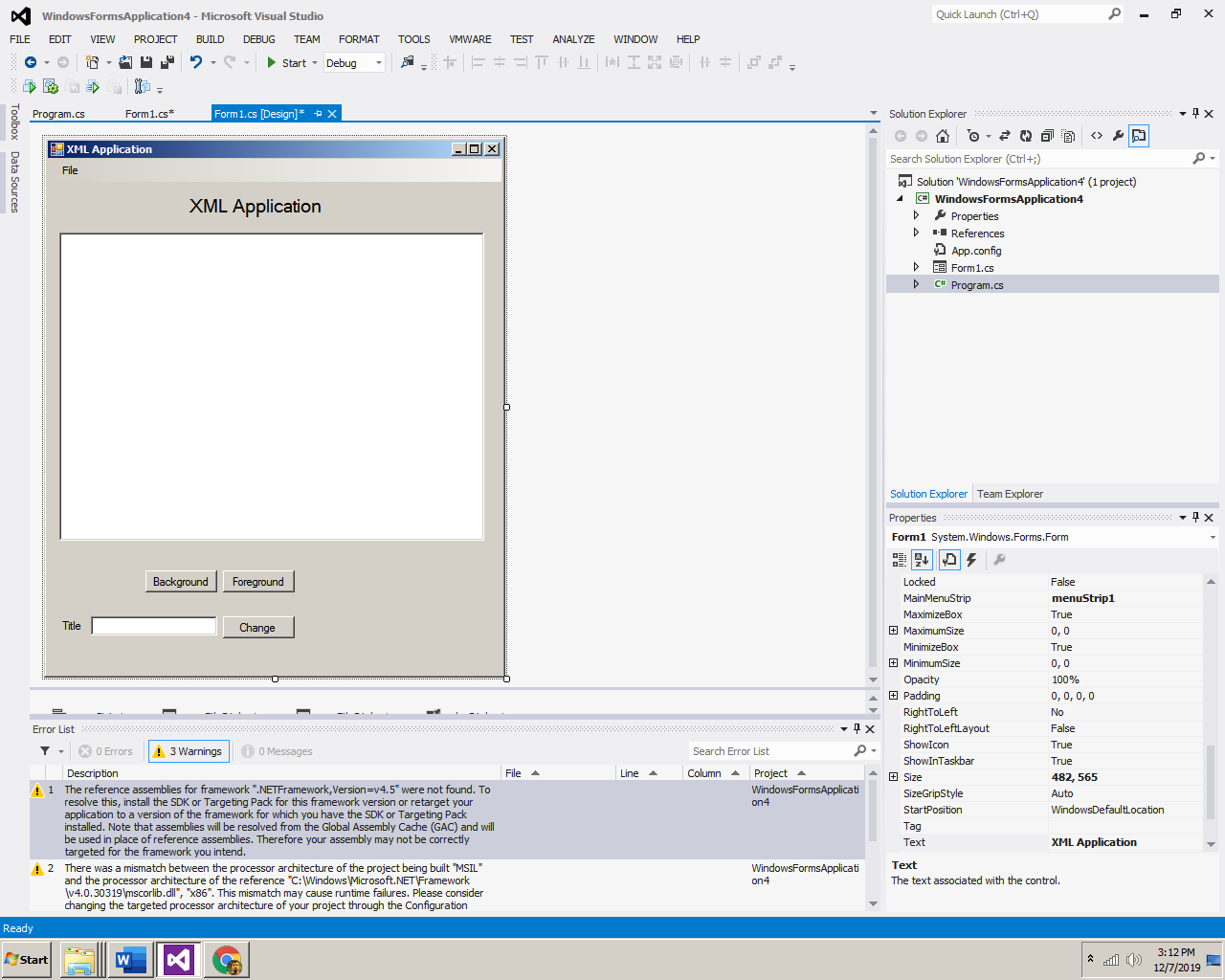
this.label1.ForeColor = Color.FromName(l[0].InnerText);

}

}

}

}

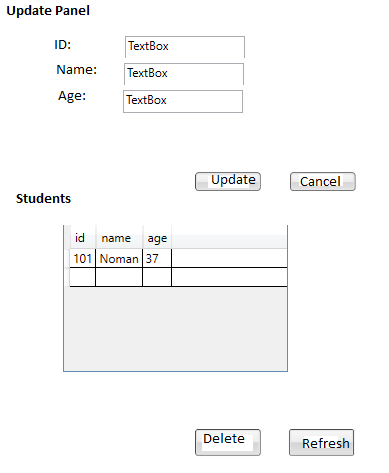


**Lab 8: To study and implement WPF and its layouts in C#**

In this lab, we will try to implement user interfaces using WPF and its layouts. We will also work with WPF data grid control.

**Lab Task:**

* 1. Design the following user interface in WPF using its various layouts.
  2. User should be able to add a new student as well as update and delete.
  3. The center data grid control should be resized when the window is maximized.



**Code**

<Window x:Class="WpfApplication1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="MainWindow" Height="350" Width="525"

Loaded ="Window\_Loaded\_1" >

<Grid>

<DockPanel>

<Grid DockPanel.Dock="Top" HorizontalAlignment="Left">

<Button Content="update" Margin="355,57,47,43" ></Button>

<Button Content="cancel" Margin="355,90,47,10"></Button>

<Label Content="id" HorizontalAlignment="Left" Margin="103,25,0,0" VerticalAlignment="Top"/>

<Label Content="age" HorizontalAlignment="Left" Margin="103,77,0,0" VerticalAlignment="Top"/>

<Label Content="name" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="103,51,0,0"/>

<TextBox HorizontalAlignment="Left" Height="23" Margin="166,51,0,0" TextWrapping="Wrap" Text="TextBox" VerticalAlignment="Top" Width="120"/>

<TextBox HorizontalAlignment="Left" Height="23" Margin="166,25,0,0" TextWrapping="Wrap" Text="TextBox" VerticalAlignment="Top" Width="120"/>

<TextBox HorizontalAlignment="Left" Margin="166,79,0,17" TextWrapping="Wrap" Text="TextBox" Width="120"/>

</Grid>

<StackPanel HorizontalAlignment="Right" DockPanel.Dock="Bottom" Orientation="Horizontal">

<Button Margin="10,0,0,0" Content="Refresh"> </Button>

<Button Margin="10,0,0,0" Content="Delete"></Button>

</StackPanel>

<DataGrid x:Name="datagrid1" HorizontalAlignment="Stretch" Margin="0,0,0,0" VerticalAlignment="Stretch" SelectionChanged="DataGrid\_SelectionChanged\_1" />

</DockPanel>

</Grid>

</Window>

**Student class**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WpfApplication1

{

class Sudent

{

public string id {set; get;}

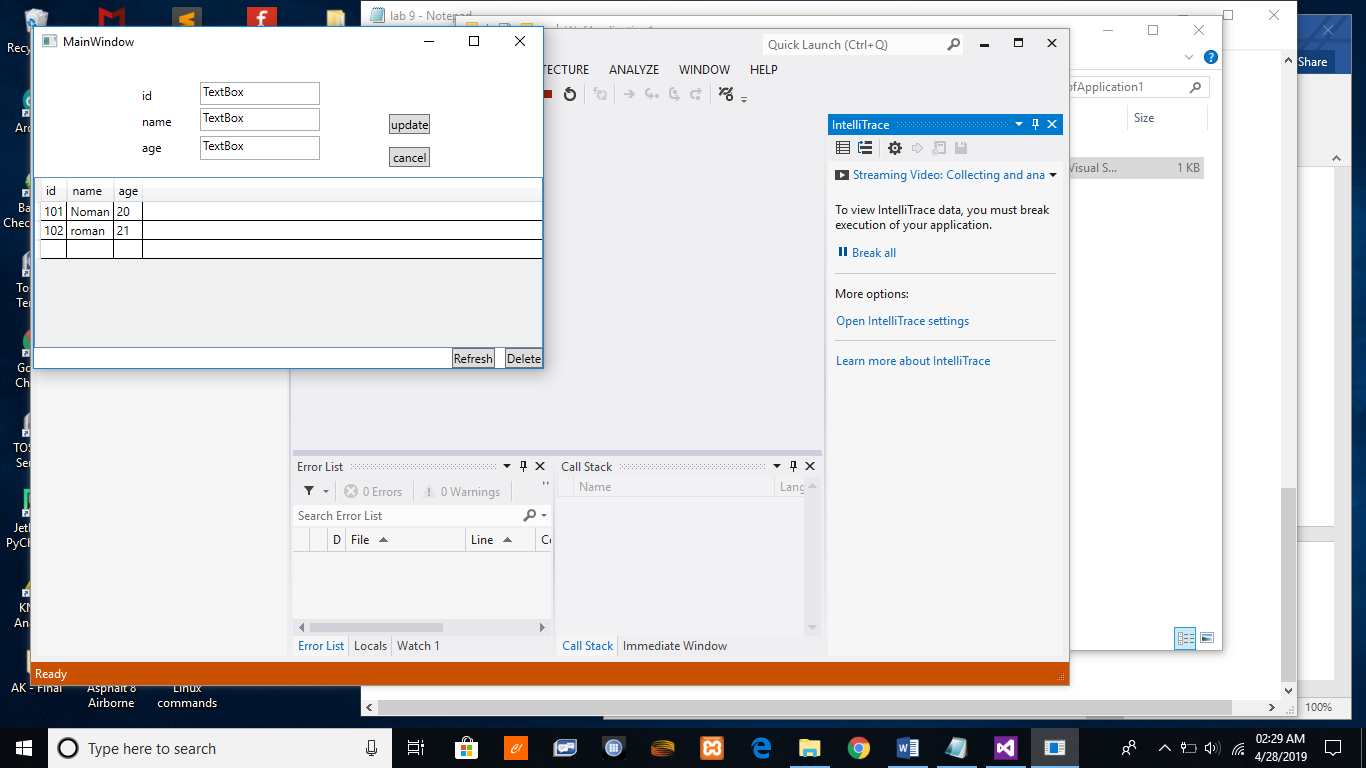
public string name {set; get;}

public string age {set; get;}

}

}

OUTPUT:



**Lab 9: To study and implement LINQ in C#**

LINQ is an extension to the C# language that integrates data query directly into the programming language itself. Visual Studio 2015 and the .NET Framework 4.5 come with a number of built-in LINQ providers that provide query solutions for different types of data

* LINQ to Objects
* LINQ to XML
* LINQ to Entities
* LINQ to Data Set

**Lab Tasks:**

**Lab Tasks:**

1. Create an array of 1000 randomly generated numbers. Use the LINQ query to find all the odd numbers from the list. Find the count of total odd numbers. Find the maximum and minimum odd number.

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication16

{

class Program

{

static void Main(string[] args)

{

int[] nums = GenerateLotsOfNumbers(100);

Console.WriteLine("Numbers");

foreach (var i in nums)

{

Console.Write(i + "");

}

Console.WriteLine("\n odd");

var result = from n in nums

where n % 2 != 0

select n;

int count = result.Count();

int max = result.Max();

int min = result.Min();

Console.WriteLine("Count {0}, Max {1}, Min {2}",count,max,min);

foreach(var i in result)

{

Console.Write(i +"");

}

Console.ReadKey();

}

private static int[] GenerateLotsOfNumbers(int count)

{

Random generator = new Random(0);

int[] result = new int[count];

for (int i = 0; i < count; i++)

{

result[i] = generator.Next();

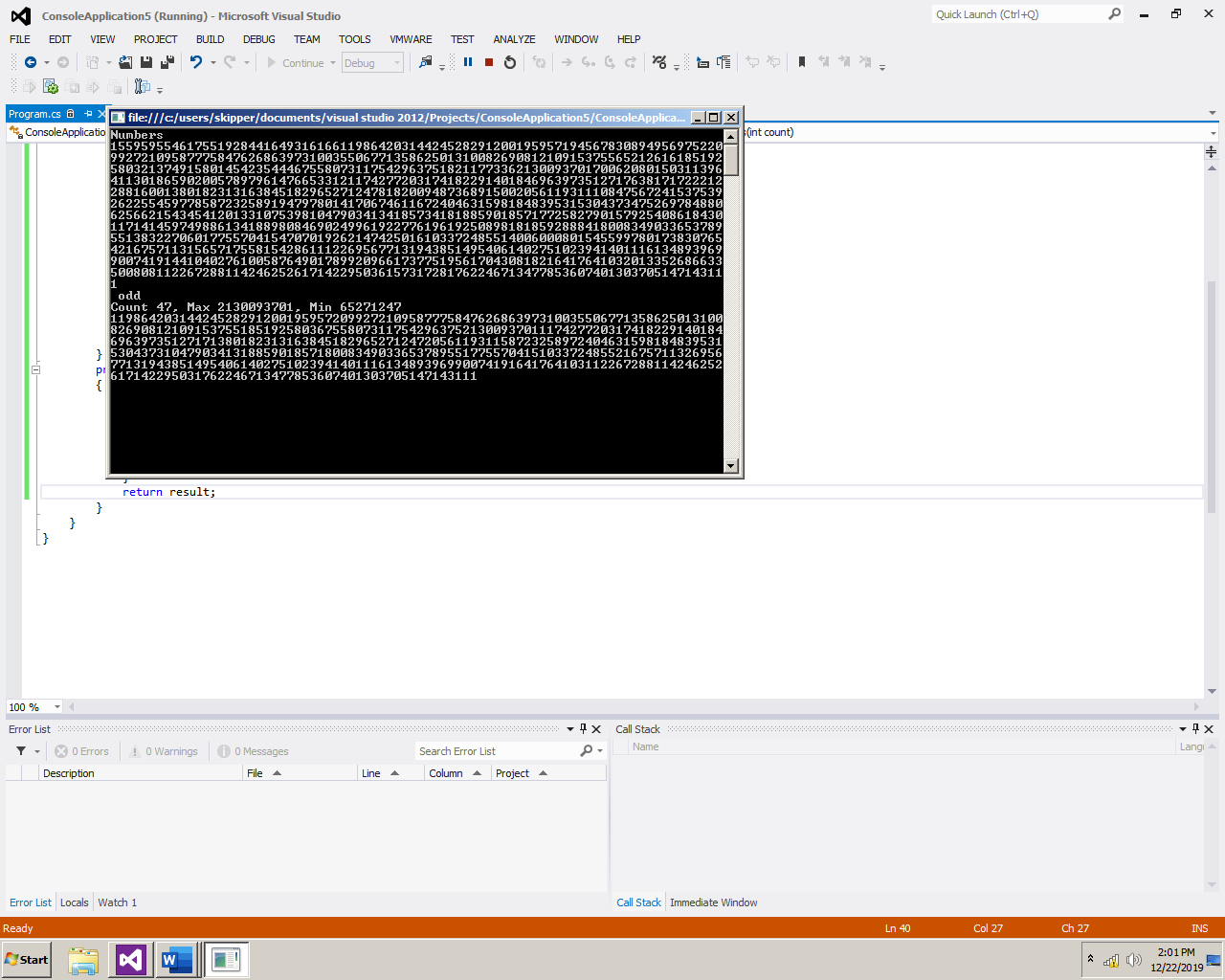
}

return result;

}

}

}



1. Create a class of Student with name, subject, and marks. Now add the students in a List. Using LINQ methods and group by, find the average marks of students.

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication17

{

class Program

{

static void Main(string[] args)

{

List<Student> stds = new List<Student>();

stds.Add(new Student ("A", "English",57));

stds.Add(new Student("B", "English", 58));

stds.Add(new Student("C", "English", 59));

var results = stds.GroupBy(s => s.name, (key, g) => new { Student = key, Average= g.Average(s=>s.marks)});

foreach(var v in results)

{

Console.WriteLine(v);

}

Console.ReadKey();

}

}

}

1. Create a WPF project in which you can add doctor’s details such as name, qualification and salary. The added information is saved in a List. Provide a text area through which user can write LINQ query that can be run against the list.

**Code:**

student class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication17

{

class Student

{

public string name { set; get; }

public string subject { set; get; }

public int marks { set; get; }

public Student(string name, string subject, int marks)

{

this.name = name;

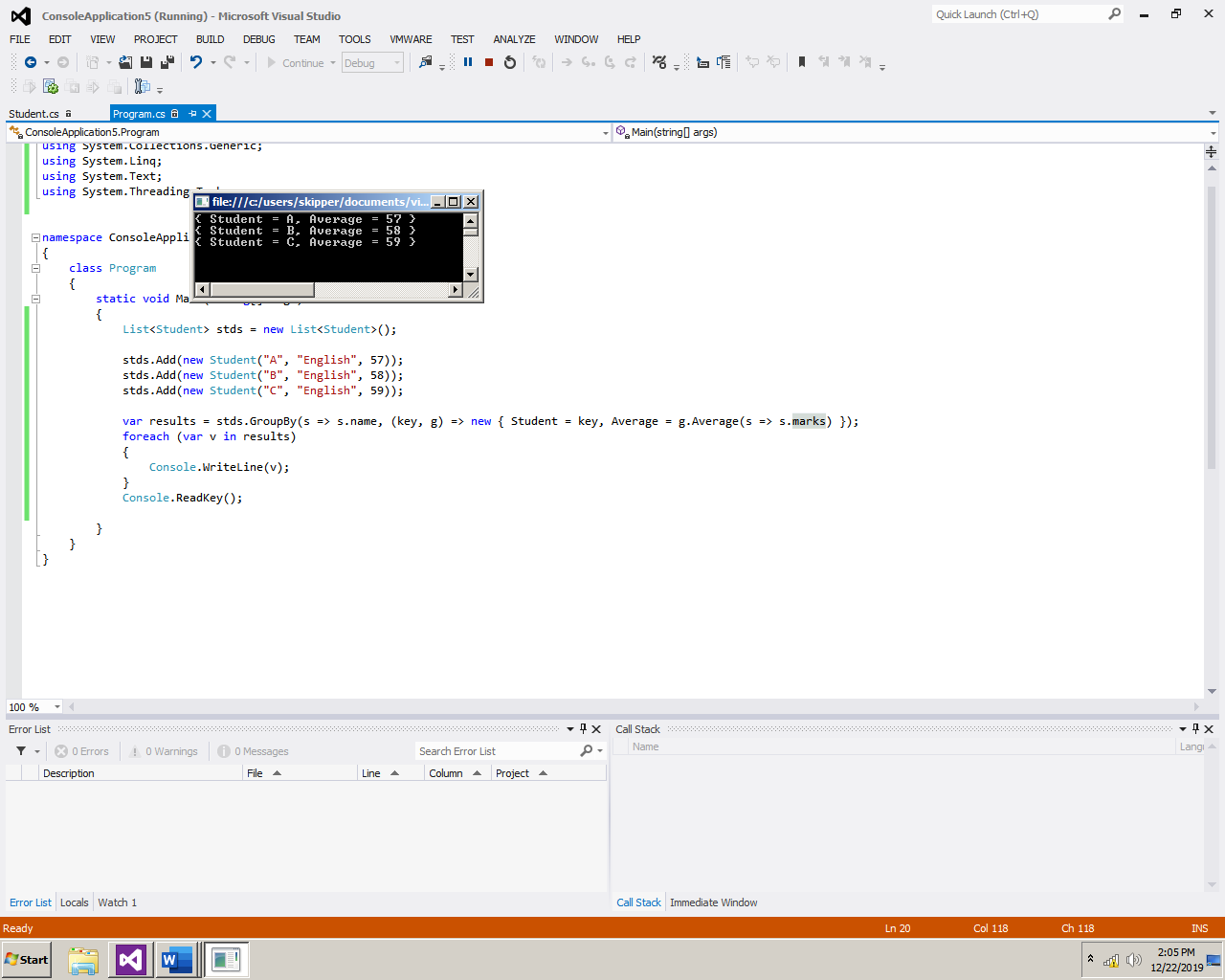
this.subject = subject;

this.marks = marks;

}

}

}



**Lab 10: To study and implement ADO .Net in C#**

The .NET platform defines a number of namespaces that allow you to interact with relational database systems. Collectively speaking, these namespaces are known as ADO.NET. In this lab, we will use ADO .net to connect to database.

**Lab Tasks:**

1. Create a database named School. Create a table Student.
2. Now create a windows form and show all the students records in a GridView
3. In the same windows form allow the user to add, update and delete a student record.

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WpfApplication2

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

List<Student> stds = new List<Student>();

public MainWindow()

{

InitializeComponent();

}

private void Window\_Loaded\_1(object sender, RoutedEventArgs e)

{

Student s;

stds.Add(s = new Student());

s.id = "101";

s.name = "Noman";

s.age = 20;

stds.Add(s = new Student());

s.id = "102";

s.name = "Saleem";

s.age = 21;

stds.Add(s = new Student());

s.id = "103";

s.name = "Najma";

s.age = 22;

datagrid1.ItemsSource = stds;

}

}

}