

Visual Programming Languages

Lab Manual

[Fall 2019]

STUDENT NAME: MUHAMMAD SANNAN

STUDENT ID: 12395

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

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **S. No** | **Date** | **Experiment** |  |
| **1** | \_\_/\_\_/\_\_ | To setup the environment and familiarize with C# |  |
| **2** | \_\_/\_\_/\_\_ | To study and understand how to write programs in C# using loops and arrays |  |
| **3** | \_\_/\_\_/\_\_ | To study and implement object oriented programming concepts in C# |  |
| **4** | \_\_/\_\_/\_\_ | To study and implement Windows Forms application in C# |  |
| **5** | \_\_/\_\_/\_\_ | To study and implement Collections in C# |  |
| **6** | \_\_/\_\_/\_\_ | To study and implement I/O in C# |  |
| **7** | \_\_/\_\_/\_\_ | To study and implement XML parsing in C# |  |
| **8** | \_\_/\_\_/\_\_ | To study and implement WPF and its layouts in C# |  |
| **9** | \_\_/\_\_/\_\_ | To study and implement LINQ in C# |  |
| **10** | \_\_/\_\_/\_\_ | To study and implement ADO .Net 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:**

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

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{ static void Main(string[] args)

{

Console.WriteLine("\n=============== CV ===============\n");

Console.WriteLine(" Name = MUHAMMAD SANNAN\n");

Console.WriteLine(" Father Name = SHAHID MAJEED \n");

Console.WriteLine(" Email Address = msannan40@gmail.com\n");

Console.WriteLine(" Career Objective:");

Console.WriteLine(" To succeed in an environment of growth and experience and earn a job \n which provides me job satisfaction and self-development and help me achieve \n personal as well as organization goals\n");

Console.WriteLine(" Educational Qualification:\n");

Console.WriteLine(" Bachelor of Science in Computer Science\n IQRA UNVIERSITY\n Gulshan-e-Iqbal\n");

Console.WriteLine(" Intermediate\n Govt. College Gulshan block 7 \n Gulshan e Iqbal, Karachi\n");

Console.WriteLine(" Matriculation\n National High School, Karachi\n");

Console.WriteLine(" Technical Skills:\n");

Console.WriteLine(" 1. Web Programming Language\n 2. Object Oriented Programming\n 3. C# \n");

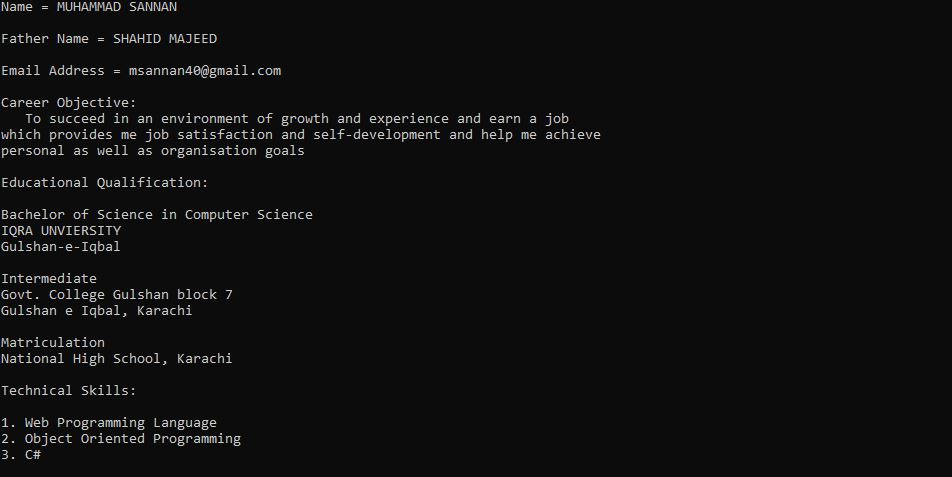
Console.ReadLine();

}

}

}

**OUTPUT**



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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int a;

Console.WriteLine("enter any number:");

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

if (a % 2 == 0)

Console.WriteLine("number is even ");

else

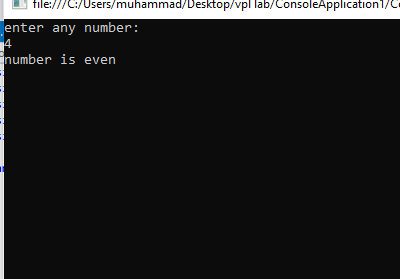
Console.WriteLine("number is ODD ");

Console.ReadKey(); }

}

}

**OUTPUT**



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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int number1;

int number2;

int number3;

Console.WriteLine("Enter your first number:");

number1 = int.Parse(Console.ReadLine());

Console.WriteLine("Enter your second number:");

number2 = int.Parse(Console.ReadLine());

Console.WriteLine("Enter your third number:");

number3 = int.Parse(Console.ReadLine());

if (number1 > number2 && number1 > number3)

{

Console.WriteLine("\nnumber 1 is greater that is " + number1);

}

else if (number2 > number3 && number2 > number1)

{

Console.WriteLine("\nnumber 2 is greater that is " + number2);

}

else if (number3 > number1 && number3 > number2)

{

Console.WriteLine("\nnumber 3 is greater that is " + number3);

}

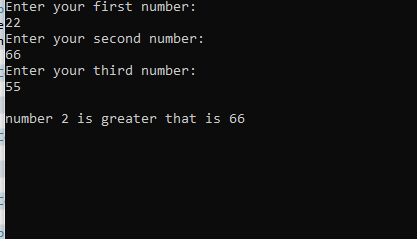
Console.ReadKey();

}

}

}

**OUTPUT**



* 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.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int month;

Console.WriteLine("Enter the month from 1-12\n");

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

if (month == 1 || month == 2 || month == 3 || month == 4)

{

Console.WriteLine("\nThe season is winter");

}

if (month == 4 || month == 5)

{

Console.WriteLine("\nThe season is summer");

}

if (month == 6 || month == 7 || month == 8)

{

Console.WriteLine("\nThe season is spring");

}

if (month == 9 || month == 10 || month == 11 || month == 12)

{

Console.WriteLine("\nThe season is autumn");

}

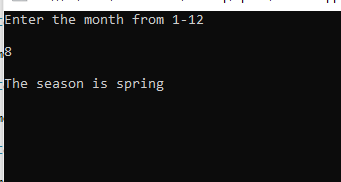
Console.ReadKey();

}

}

}

**OUTPUT**



* 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 a year as integer. Using if…else, determines whether the input is a leap year or not.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

{

int year;

Console.WriteLine("Enter the year\n");

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

if (year % 4 == 0)

{

if (year % 100 == 0)

{

Console.WriteLine("\nIt is a leap year");

}

else

Console.WriteLine("\nIt is not a leap year");

}

if (year % 4 == 0)

{

if (year % 100 == 0)

{

if (year % 400 == 0)

{

Console.WriteLine("\nIt is a leap year");

}

}

}

else

Console.WriteLine("\nIt is not a leap year");

}

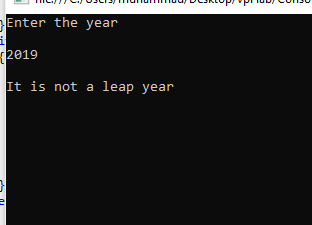
Console.ReadKey();

}

}

}

**OUTPUT**



* 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.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int firstnum;

int secodnum;

int total;

string op;

Console.WriteLine("Enter your first number:");

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

Console.WriteLine("\nEnter your second number:");

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

Console.WriteLine("\nEnter your operator:");

op = Console.ReadLine();

switch (op)

{

case "+": total = firstnum + secodnum;

Console.WriteLine("\nResult = " + total);

break;

case "-": total = firstnum - secodnum;

Console.WriteLine("\nResult = " + total);

break;

case "\*": total = firstnum \* secodnum;

Console.WriteLine("\nResult = " + total);

break;

case "/": total = firstnum / secodnum;

Console.WriteLine("\nResult = " + total);

break;

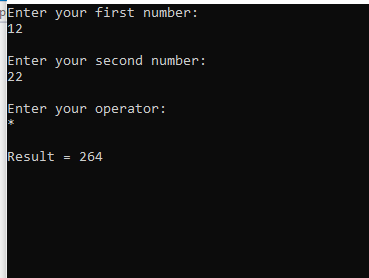
}

Console.ReadKey();

}

}

}

**OUTPUT**  


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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

{

float firstnum;

float secodnum;

float thirdnum;

float fourthnum;

float fifthnum;

float total;

float PER;

Console.WriteLine("Enter your marks in WPL: ");

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

Console.WriteLine("\nEnter your marks in OS: ");

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

Console.WriteLine("\nEnter your marks in DAA: ");

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

Console.WriteLine("\nEnter your marks in ML: ");

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

Console.WriteLine("\nEnter your marks in VPL: ");

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

Console.WriteLine("\nMarksheet:\n");

Console.WriteLine(" WPL: " + firstnum);

Console.WriteLine(" OS: " + secodnum);

Console.WriteLine(" DAA: " + thirdnum);

Console.WriteLine(" ML: " + fourthnum);

Console.WriteLine(" VPL: " + fifthnum);

total = firstnum + secodnum + thirdnum + fourthnum + fifthnum;

Console.WriteLine("\nTOTAL MARKS " + total + " OUT OF 500");

PER = (total / 500) \* 100;

Console.WriteLine("\nPERCENTAGE = " + PER);

}

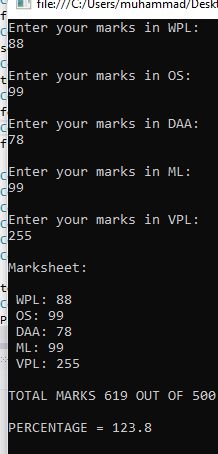
Console.ReadKey();

}

}

}

**Output**

****

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

**Tasks:**

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

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication2

{

class Program

{

static void Main(string[] args)

{

int i = 0;

int count = 0;

int element = 0;

int[] arr1 = new int[5];

Console.WriteLine("Enter numbers : ");

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

{

Console.Write("\nElement[" + (i + 1) + "]: ");

arr1[i] = int.Parse(Console.ReadLine());

}

Console. Write("\nEnter Element Number : ");

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

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

{

if (element == arr1[i])

{

count++;

}

}

Console.WriteLine("\nTotal occurrence of number " + element + " is : " + count);

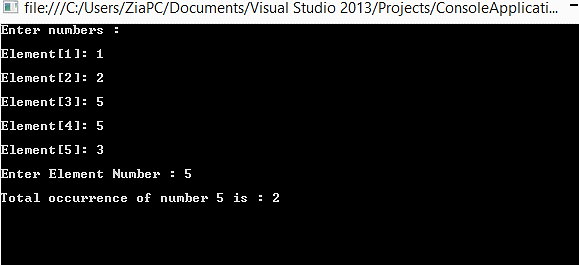
Console.ReadLine();

}

}

}

**Output:**



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

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication2

{

class Program

{

static void Main(string[] args)

{

int max;

int min;

int[] a = { 4, 6, 9, 14, 12, 2 };

max = a[0];

min = a[0];

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

{

if (a[i]>max)

{

max = a[i];

}

if(a[i]<min)

{

min = a[i];

}

}

Console.WriteLine("Maximum is "+max);

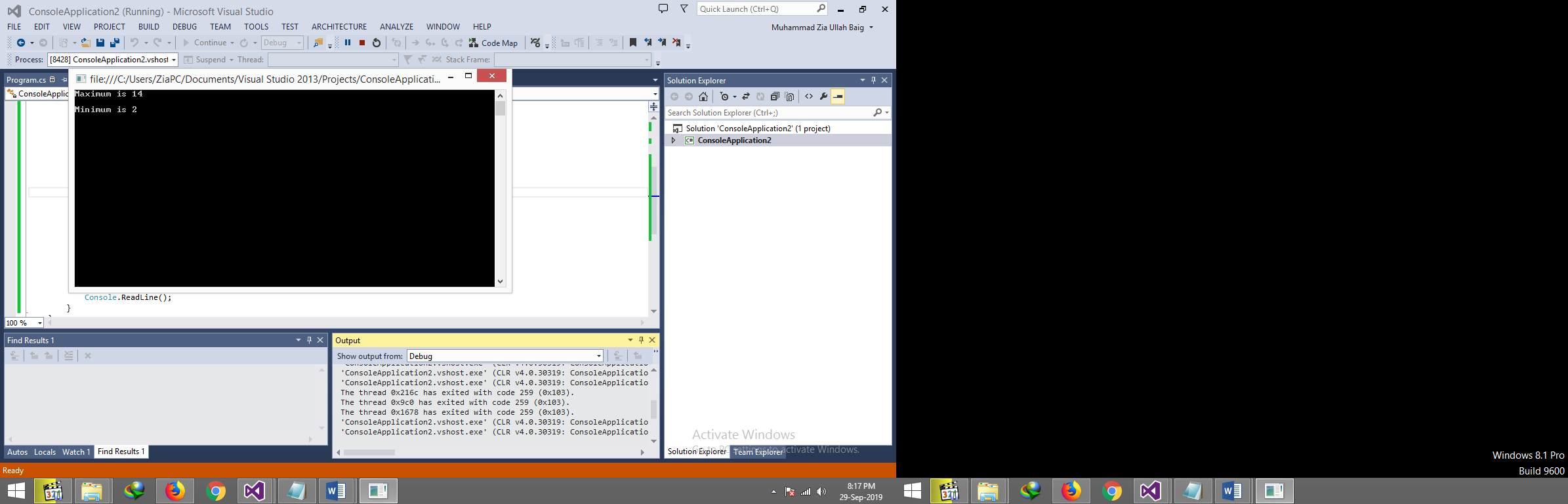
Console.WriteLine("\nMinimum is "+min);

Console.ReadLine();

}

}

}

**Output:**

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

using System;  
using System.Collections;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;

namespace ConsoleApplication2

{

class Program

{

static void Main(string[] args)

{

int[] x = { 3, 8, 5, 12, 17, 2 };

ArrayList even = new ArrayList();

ArrayList odd = new ArrayList();

foreach(int v in x)

{

if (v % 2 == 0)

{

even.Add(v);

}

else

{

odd.Add(v);

}

}

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

foreach (int i in even)

{

Console.Write("\n {0} \n", i);

}

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

foreach (int j in odd)

{

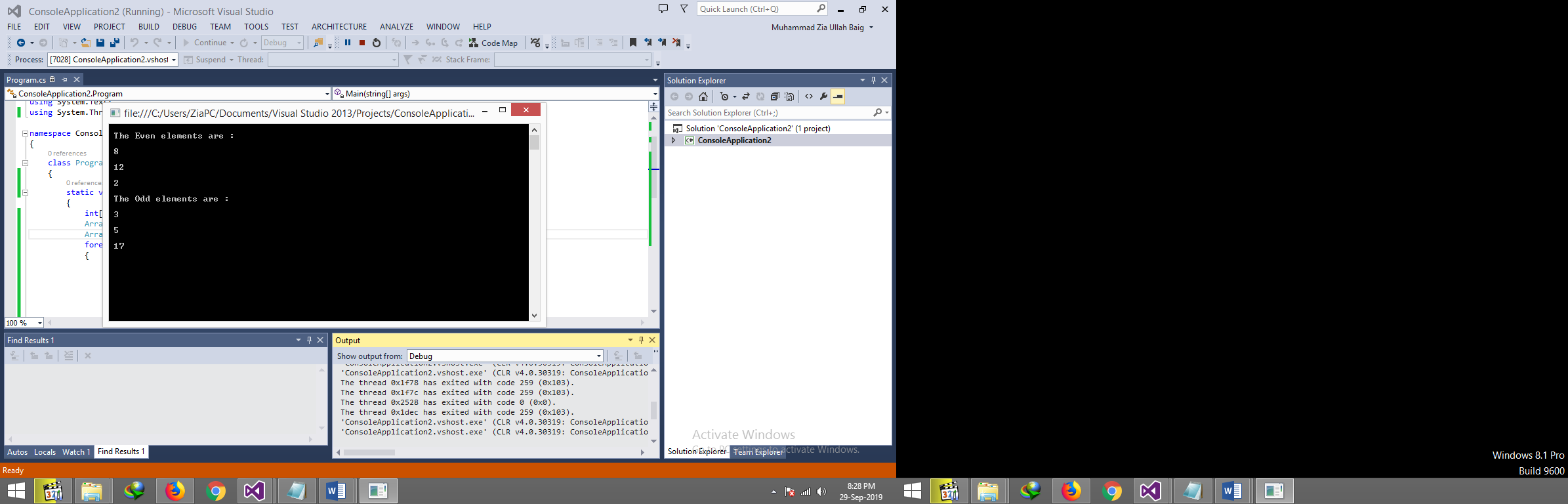
Console.Write("\n {0} \n", j);

}

Console.ReadLine();

}

}  
}

**Output:**

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

**Code:**using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication2

{

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("\nlength of the string is: {0}\n", length);

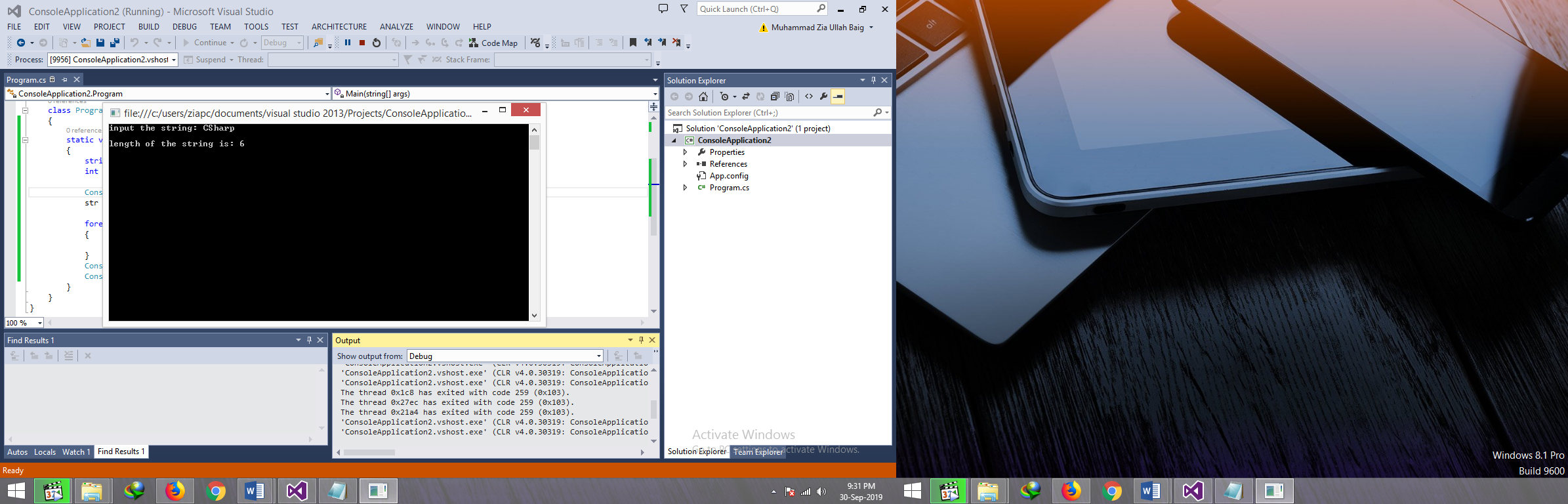
Console.ReadLine();

}

}

}

**Output:**



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

**Code:**using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication2

{

class Program

{

static void Main(string[] args)

{

string str;

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

str = Console.ReadLine();

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

foreach (string word in words)

{

Console.WriteLine("\n WORD: " + word);

}

Console.WriteLine("\n Total number of words");

Console.WriteLine("\n "+ words.Length);

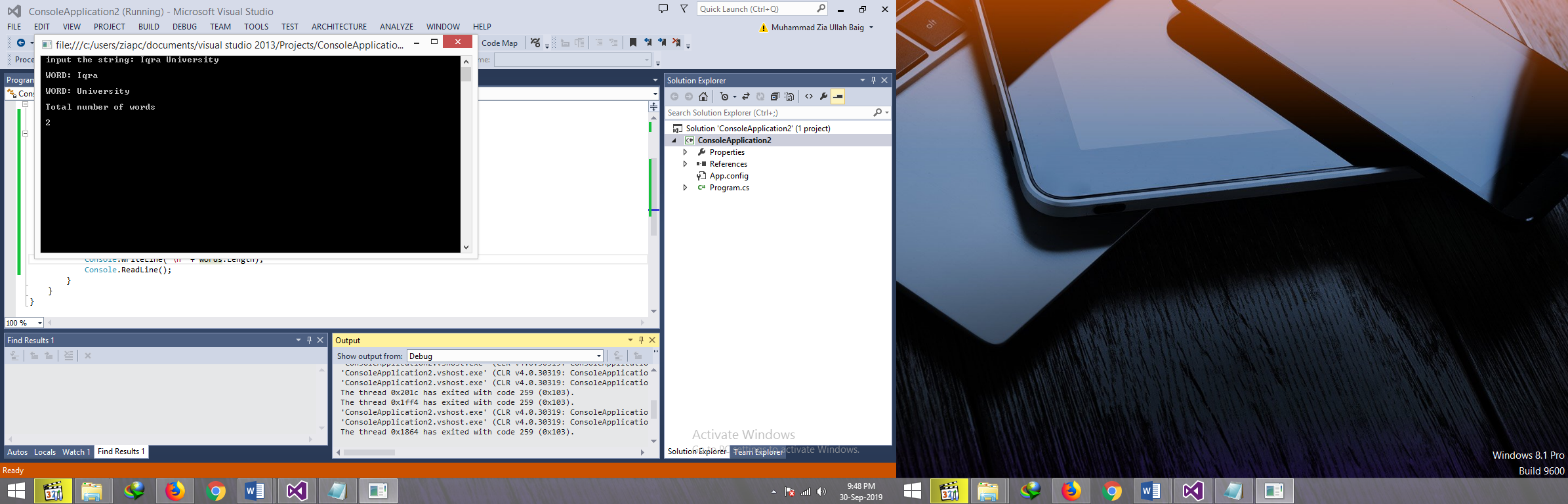
Console.ReadLine();

}

}

}

**Output:**



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

**Code:**using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication2

{

class Program

{

static int Main(string[] args)

{

int n, i = 0, c;

Console.WriteLine("Enter the number of terms:\n");

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

Console.WriteLine("\nFibonacci series\n");

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

{

int result = FibonacciFunction(i);

Console.Write(result + " ");

i++;

}

Console.ReadLine();

return 0;

}

public static int FibonacciFunction(int n)

{

if (n == 0)

{

return 0;

}

else if (n == 1)

{

return 1;

}

else

{

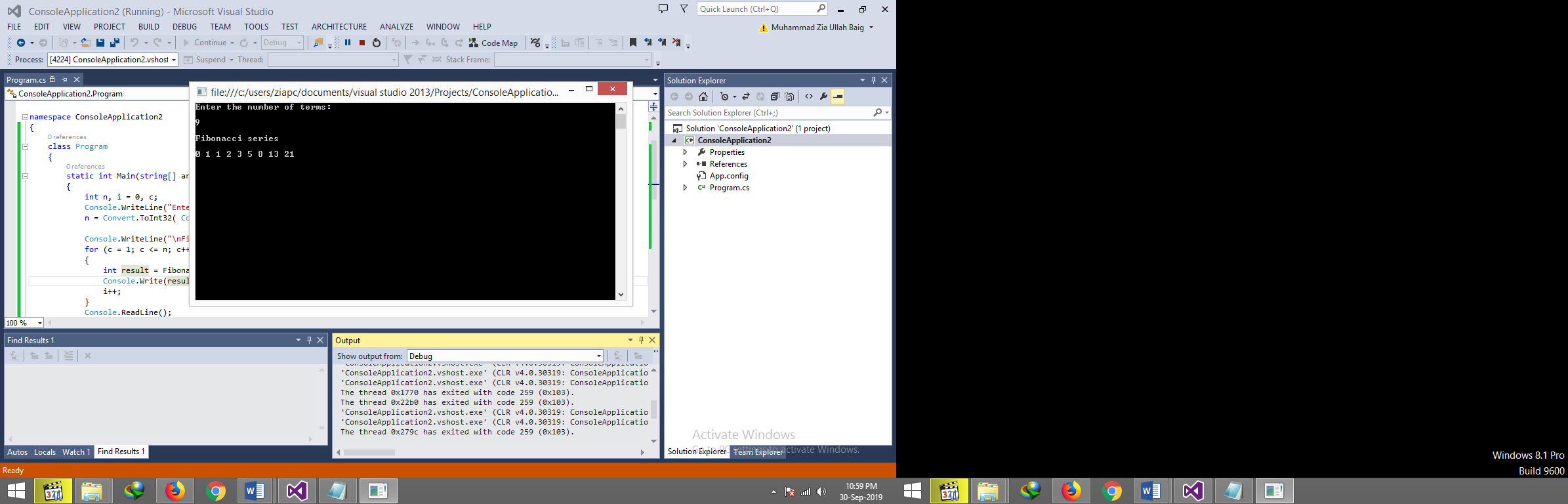
return (FibonacciFunction(n - 1) + FibonacciFunction(n - 2));

}

}

}

}

**Output:**

**Lab 3: To study and implement object oriented programming**

**concepts in C#**

**Task 1:**

1. Encapsulation
   * + 1. Create a class Circle.
       2. Define a property named radius. In the set method, check if the radius is negative then throw an exception System.ArgumentException.
       3. Define a one argument constructor to initialize the radius.
       4. Define a no argument constructor to initialize the radius value to zero using constructor chaining.
       5. Define a method GetArea() to calculate the area of circle.
       6. Create a Test class.
       7. In the Main method, define two objects of Circle and initialize them with random values.
       8. Call the GetArea() method of each object and print the area.

**Circle Class Code:**using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication3

{

class Circle

{

private float r;

//define property of radius

public float radius

{

get

{

Console.WriteLine(" Getter called");

return r;

}

set

{

if (this.r < 0)

{

throw new ArgumentException(" Negative value not allowed");

}

else

{

this.r = value;

}

}

}

//Constructor with one argument

public Circle(float rad)

{

this.r = rad;

}

//Constructor with no arguments

public Circle()

: this(0)

{

}

//GetArea() method to calculate area

public void GetArea()

{

double a = Math.PI \* r \* r;

Console.WriteLine(a);

}

}

}

**Main Method:**using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication3

{

class Program

{

static void Main(string[] args)

{

Circle c = new Circle(10);

Circle cc = new Circle(12);

// Call getarea method

Console.WriteLine("\nFor First Object the area is: ");

c.GetArea();

//call getarea method for second object

Console.WriteLine("\nFor Second Object the area is: ");

cc.GetArea();

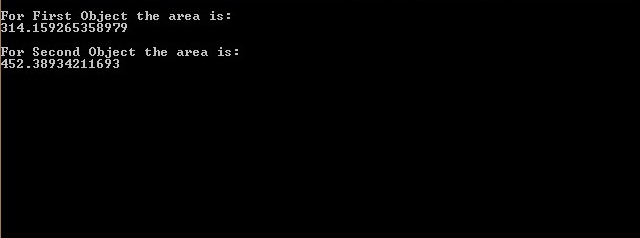
Console.ReadLine();

}

}

}

**Output:**

****

**Task 2:**

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

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

* + - 1. **Abstract Class Animal Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication3

{

public abstract class Animal

{

private string namee;

//property name type string

public string name

{

get { return this.namee; }

set { this.namee = value; }

}

//abstract method sound()  
 public abstract void sound();

//Constructor to initialize animal name

public Animal(string name)

{

this.namee = name;

}

}

}

* + - 1. **Two abstract classes**

**Abstract Class Mammal Code:**using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication3

{

//Mammal class that inherits Animal class

public abstract class Mammal : Animal

{

}

}

**Abstract Class Non-Mammal Code:**using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication3

{

//Non-Mammal class that inherits Animal class

public abstract class Non\_Mammal : Animal

{

}

}

* + - 1. **Cat Class Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication3

{

//inherit Mammal class to define Cat class

class Cat : Mammal

{

//method that prints sound

public override void Sound()

{

Console.WriteLine("Meow");

}

}

}

* + - 1. **Goat Class Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication3

{

class Goat : Mammal

{

public override void Sound()

{

Console.WriteLine("meeeee");

}

}

}

**Fish Class Code:**using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication3

{

class Fish : Non\_Mammal

{

public override void Sound()

{

Console.WriteLine("kansdkadhaojdi");

}

}

}

**Main Method:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace inheritance

{

class Program

{

static void Main(string[] args)

{

cat obj = new cat();

obj.sound();

goat obj2 = new goat();

obj2.sound();

fish obj3 = new fish();

obj3.sound();

}

}

public abstract class animal {

string name;

public string namee

{

get { return name; }

set { name = value; }

}

public abstract void sound();

public animal(string name)

{

this.name = name;

}

public animal() {

}

}

public abstract class mamal: animal

{

}

public abstract class nonmamal : animal

{

}

public class cat : mamal {

public override void sound()

{

Console.WriteLine("Cat sound : Meow");

}

}

public class goat : mamal {

public override void sound()

{

Console.WriteLine("Goat sound: meeeeeee");

}

}

public class fish : nonmamal {

public override void sound()

{

Console.WriteLine("Fish sound : hdkjasfkdgf");

Console.ReadLine();

}

}

}

**Output:**

****

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

**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.

* Design a basic Sign Up form with fields name, id, password, email, address, gender and date of birth.
* Add a checkbox to accept license agreement, and a view button to view the license agreement.
* Add a Sign Up button.
* 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

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 signup

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

}

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

{

}

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

{

string name = txtname.Text;

string id = this.txtid.Text;

string email = this.txtemail.Text;

string dob = this.dateTimePicker1.Text;

string pass = this.txtpass.Text;

string gender = "Male";

if (this.female.Checked)

{

gender = "female";

}

else if(this.male.Checked){

gender = "male";

}

bool license = this.checkLic.Checked;

if(name == "" ||

id == "" ||

email == ""||

dob == ""||

pass == "" ||

license == false )

{

MessageBox.Show("please fill all the box");

return;

}

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

!email.Contains("."))

{

MessageBox.Show("please provide a valid email");

return;

}

}

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

{

}

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

{

Form2 obj = new Form2();

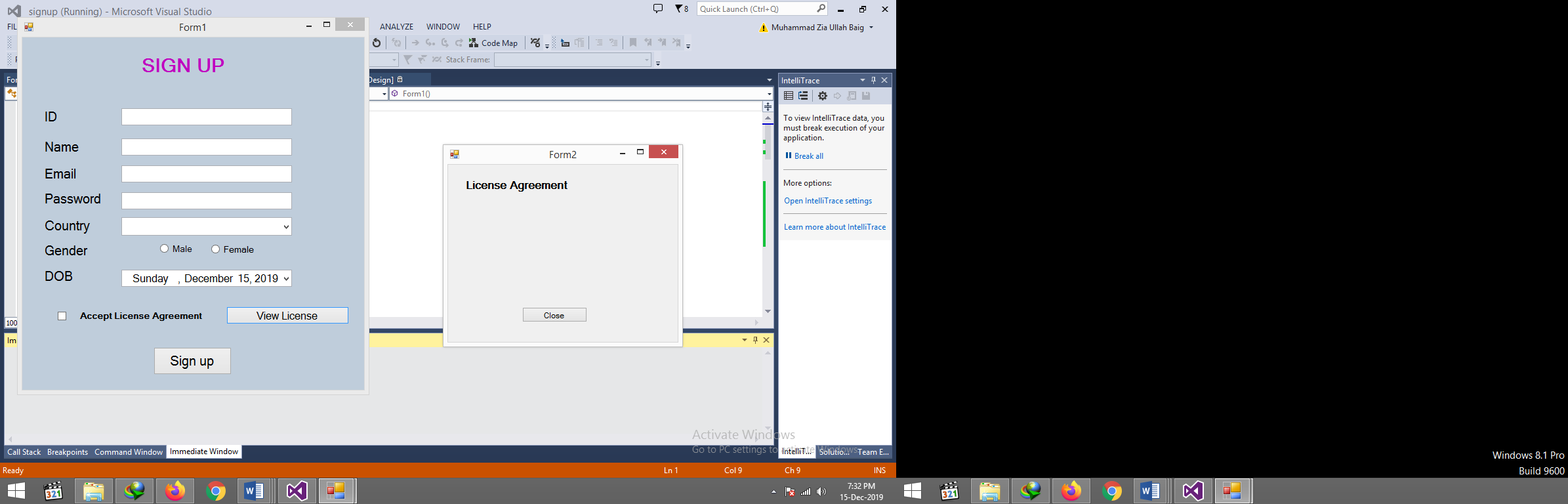
obj.Show();

}

}

}

**Output:**



* 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.Threading.Tasks;

using System.Windows.Forms;

namespace signup

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

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

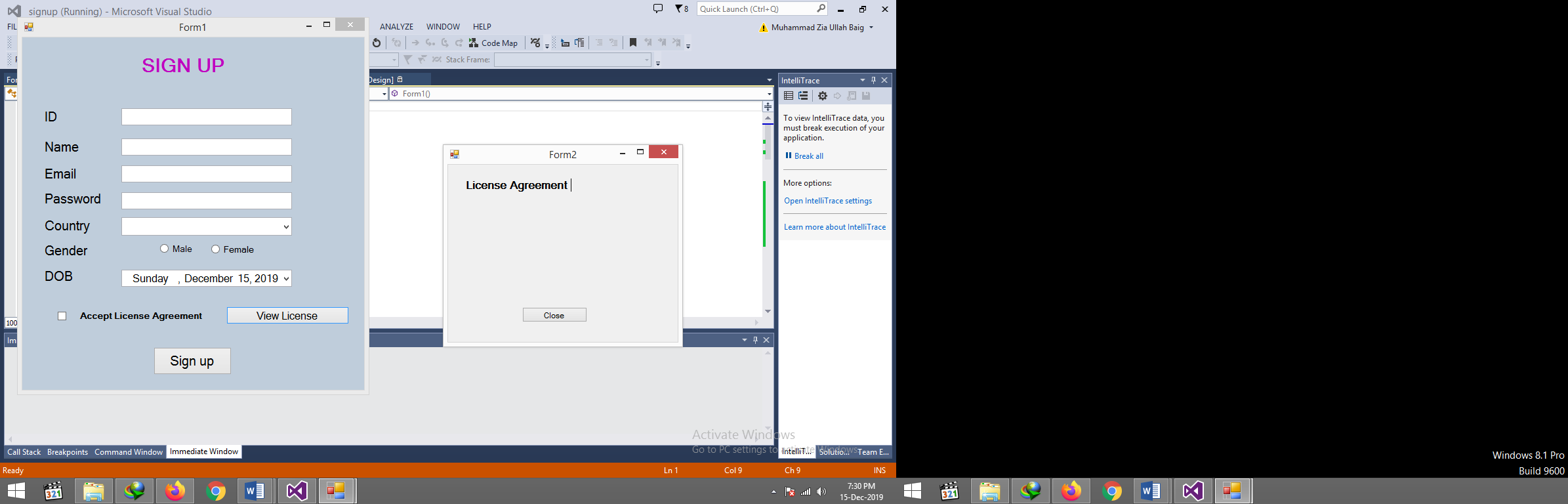
{

this.Close();

}

}

}

**Output:**

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

**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.

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();

}

}

}

**Class Student:**

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;

}

}

}

**AddStudent:**

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)

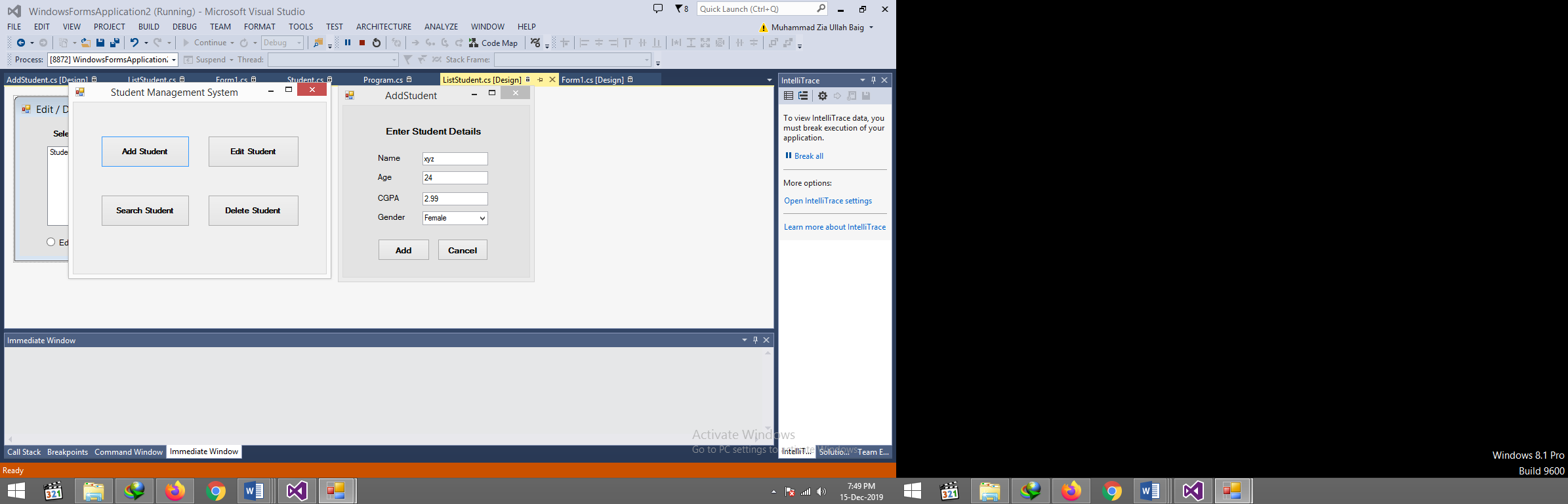
{

}

}

}

**OUTPUT**



**List Student:**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();

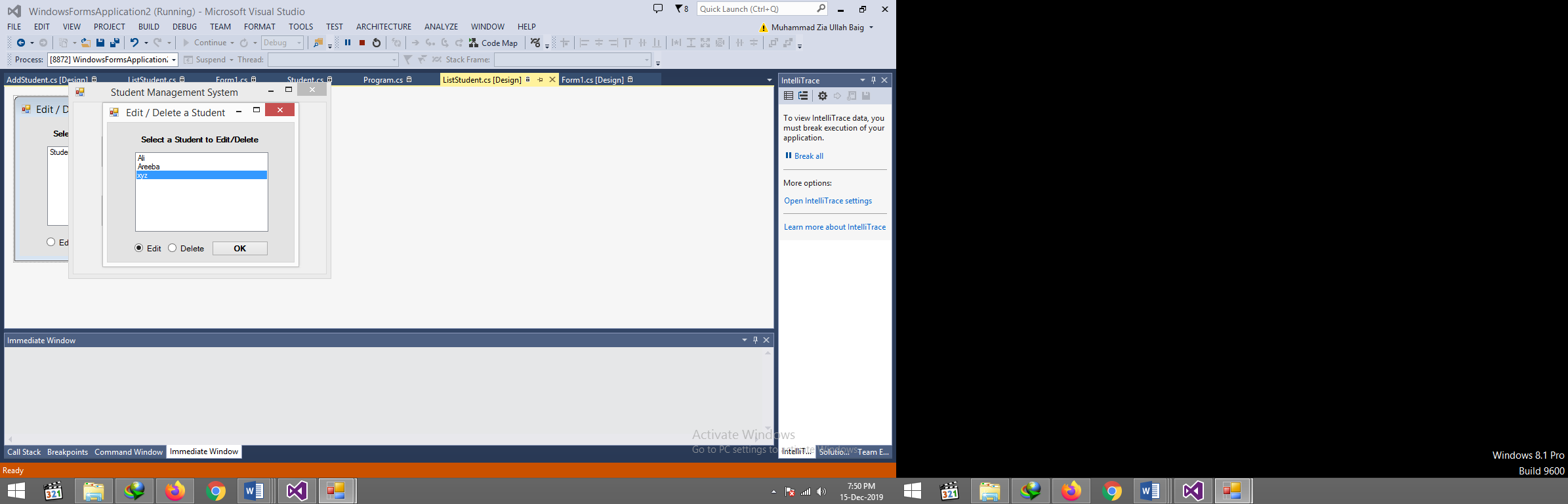
}

}

}

}

**Output:**



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

**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.

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.IO;

namespace lab6

{

public partial class Form1 : Form

{

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

int current =-1;

public Form1()

{

InitializeComponent();

}

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

{

}

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

{

Student std = new Student();

std.id = textBox1.Text;

std.age = textBox2.Text;

if (Male.Checked)

{

std.gender = "male";

}

else {

std.gender = "female";

}

std.marks = textBox3.Text;

students.Add(std);

current = students.Count - 1;

MessageBox.Show("Student added successfully");

}

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

{

Student s = students[current];

s.id = textBox1.Text;

s.age = textBox2.Text;

s.marks = textBox3.Text;

if (Male.Checked)

{

s.gender = "Male";

}

else {

s.gender = "Female";

}

}

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

{

students.RemoveAt(current);

current--;

}

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

{

}

private void eXITToolStripMenuItem1\_Click(object sender, EventArgs e)

{

this.Close();

}

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

{

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

{

StreamWriter sw = new StreamWriter(saveFileDialog1.FileName,true);

foreach (Student s in students)

{

sw.WriteLine(s.id + " " + s.age + " " + s.marks+ " "+s.gender);

}

sw.Close();

MessageBox.Show("File Saved Successfully");

}

}

private void saveFileDialog1\_FileOk(object sender, CancelEventArgs e)

{

}

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

{

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

{

StreamReader sr = new StreamReader(openFileDialog1.FileName,true);

string line = sr.ReadLine();

students = new List<Student>();

while (line != null)

{

string[] tokens = line.Split();

Student s = new Student();

s.id = tokens[0];

s.age = tokens[1];

s.marks = tokens[2];

s.gender = tokens[3];

students.Add(s);

line = sr.ReadLine();

}

current = 0;

}

}

}

}

**Class Student:**using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab6

{

class Student

{

public string id;

public string age;

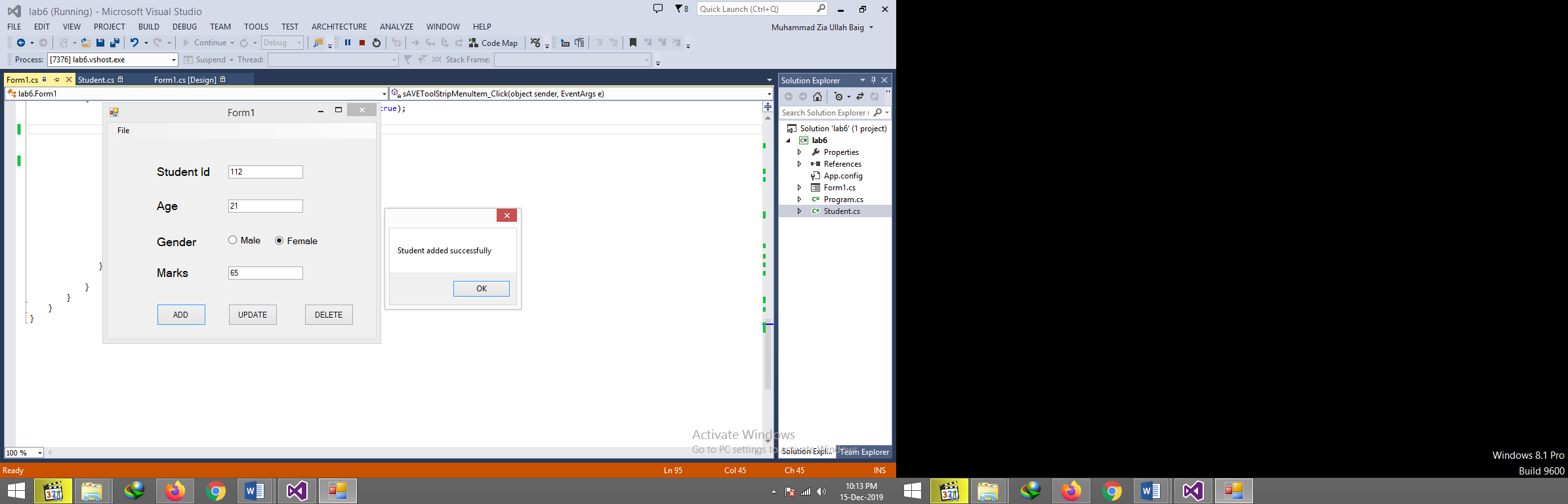
public string gender;

public string marks;

}

}

**Output:**



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

**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.

**XML Code:**

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<background>blue</background>

<foreground>black</foreground>

<text>Extensible Markup language (XML)</text>

<title>VPL</title>

<width>100</width>

<height>100</height>

</configuration>

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 WindowsFormsApplication6

{

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)

{

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);

}

}

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)

{

}

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

{

}

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

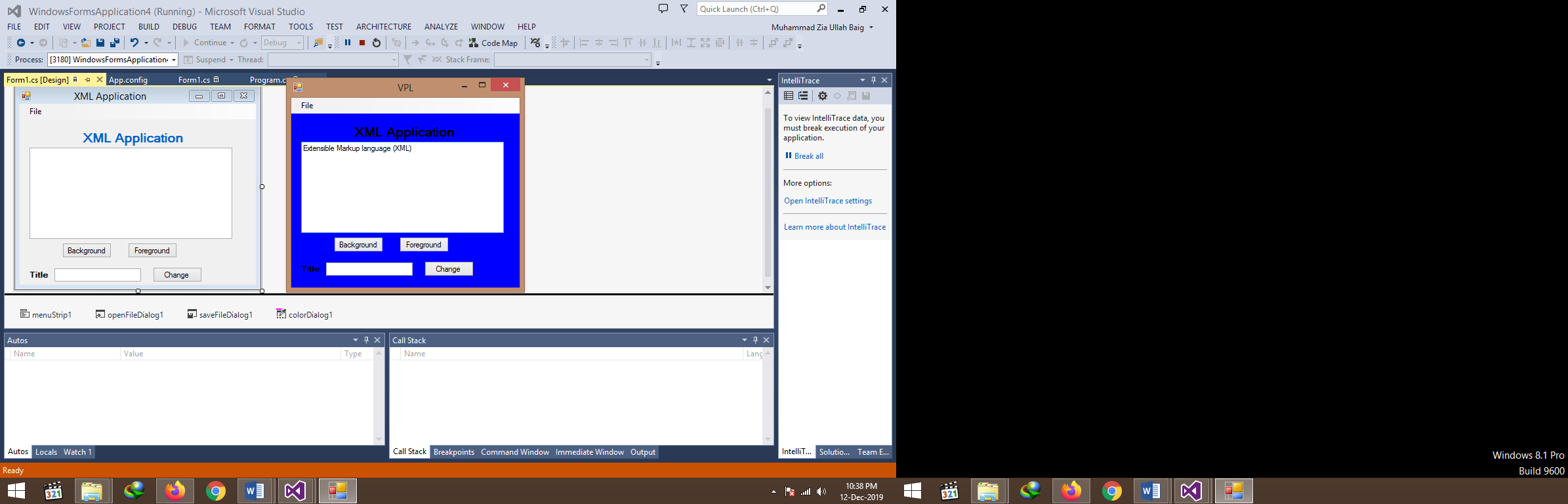
{

}

}

}

**Output:**



**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.

<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>

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

{

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 = "Saleem";

s.age = 21;

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

s.id = "102";

s.name = "Najma";

s.age = 22;

datagrid1.ItemsSource = stds;

}

}

}

**Class Student:**using System;using System.Collections.Generic;using System.Linq;using System.Text;using System.Threading.Tasks;

namespace WpfApplication2

{

class Student

{

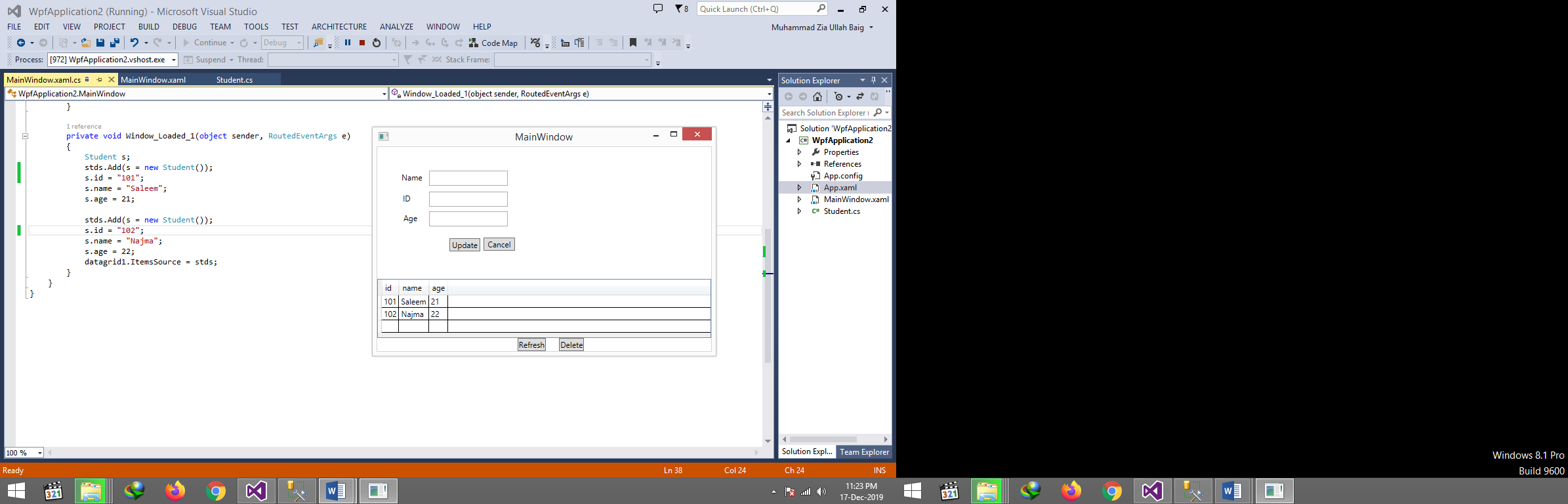
public string id { set; get; }

public string name { set; get; }

public int 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:**

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.
2. 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.
3. 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.

**1)**

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;

}

}

}

**2)**

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();

}

}

}

**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 GridViewIn the same windows form allow the user to add, update and delete a student record.

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

{

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;

}

}

}

**Output**

