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

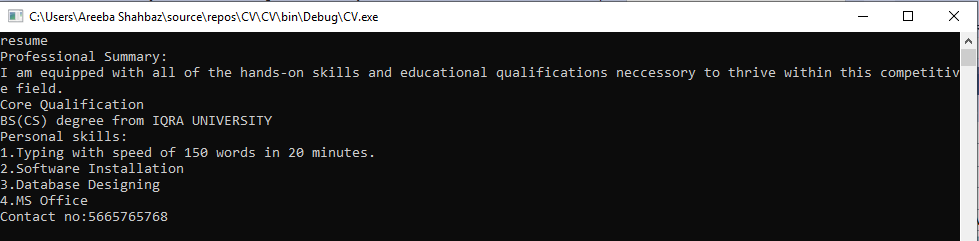
The objective of this lab is to set up the Visual Studio environment and get some familiarity with the C# language.

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.
2. using System;
3. using System.Collections.Generic;
4. using System.Linq;
5. using System.Text;
6. namespace ConsoleApplication1
7. {
8. class Program
9. {
10. static void Main(string[] args)
11. {
12. using System;
13. using System.Collections.Generic;
14. using System.Linq;
15. using System.Text;
16. using System.Threading.Tasks;
17. namespace CV
18. {
19. class Program
20. {
21. static void Main(string[] args)
22. {
23. Console.WriteLine("resume");
24. Console.WriteLine("Professional Summary:");
26. Console.WriteLine("I am equipped with all of the hands-on skills and educational qualifications neccessory to thrive within this competitive field.");
28. Console.WriteLine("Core Qualification");
30. Console.WriteLine("BS(CS) degree from IQRA UNIVERSITY");
32. Console.WriteLine("Personal skills:");
34. Console.WriteLine("1.Typing with speed of 150 words in 20 minutes.");
36. Console.WriteLine("2.Software Installation");
38. Console.WriteLine("3.Database Designing");
40. Console.WriteLine("4.MS Office");
42. Console.WriteLine("Contact no:5665765768");
44. Console.WriteLine("");
45. Console.ReadLine();
46. }
47. }
48. }

}



2.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;

namespace ConsoleApplication2

{

class Program

{

static void Main(string[] args)

{

int i;

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

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

if (i % 2 == 0)

{

Console.Write("Entered Number is an Even Number");

Console.Read();

}

else

{

Console.Write("Entered Number is an Odd Number");

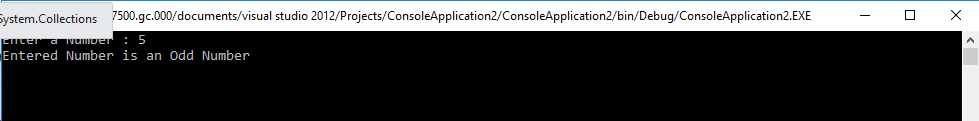
Console.Read();

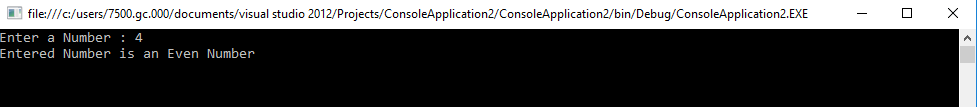
}

}

}

}





3.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;

namespace ConsoleApplication3

{

class Program

{

static void Main(string[] args)

{

int num1, num2, num3;

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

Console.Write("Find the largest of three numbers:\n");

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

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

Console.Write("Input the 1st number :");

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

Console.Write("Input the 2nd number :");

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

Console.Write("Input the 3rd number :");

num3 = Convert.ToInt32(Console.ReadLine());

if (num1 > num2)

{

if (num1 > num3)

{

Console.Write("The 1st Number is the greatest among three. \n\n");

Console.ReadLine();

}

else

{

Console.Write("The 3rd Number is the greatest among three. \n\n");

Console.ReadLine();

}

}

else if (num2 > num3)

{

Console.Write("The 2nd Number is the greatest among three \n\n");

Console.ReadLine();

}

else

{

Console.Write("The 3rd Number is the greatest among three \n\n");

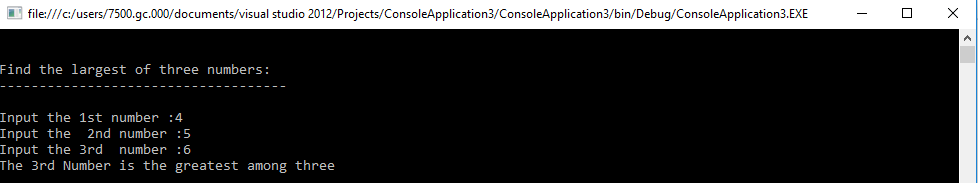
Console.ReadLine();

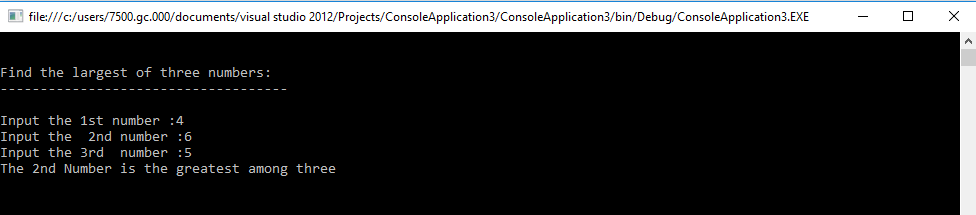
}

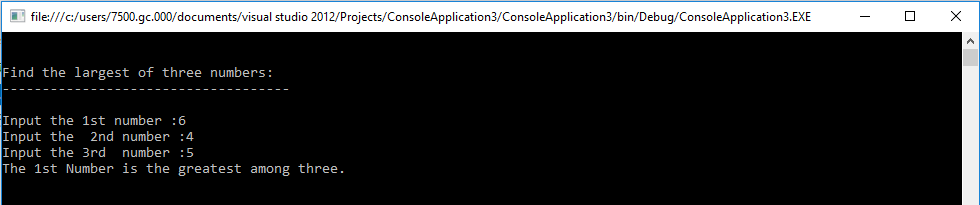
}

}

}







4.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;

namespace ConsoleApplication4

{

class Program

{

static void Main(string[] args)

{

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("Winter\n");

Console.ReadLine();

break;

case 2:

Console.Write("Winter\n");

Console.ReadLine();

break;

case 3:

Console.Write("Spring\n");

Console.ReadLine();

break;

case 4:

Console.Write("Spring\n");

Console.ReadLine();

break;

case 5:

Console.Write("Spring\n");

Console.ReadLine();

break;

case 6:

Console.Write("Summer\n");

Console.ReadLine();

break;

case 7:

Console.Write("Summer\n");

Console.ReadLine();

break;

case 8:

Console.Write("Summer\n");

Console.ReadLine();

break;

case 9:

Console.Write("Fall\n");

Console.ReadLine();

break;

case 10:

Console.Write("Fall\n");

Console.ReadLine();

break;

case 11:

Console.Write("Fall\n");

Console.ReadLine();

break;

case 12:

Console.Write("Winter\n");

Console.ReadLine();

break;

default:

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

Console.ReadLine();

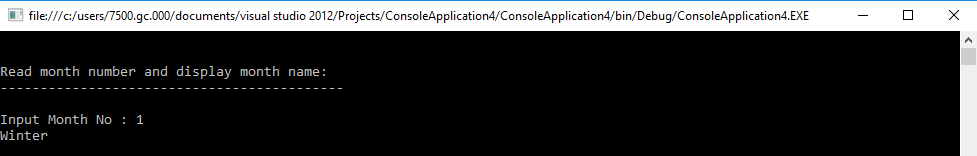
break;

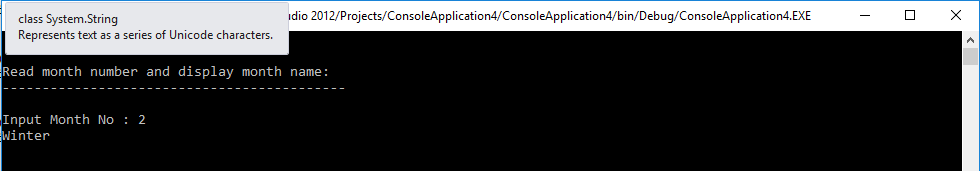
}

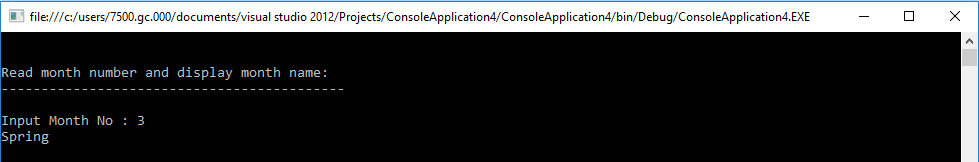
}

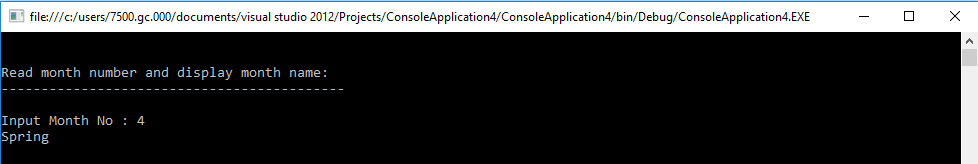
}

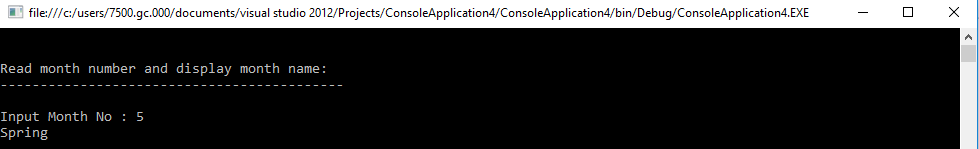
}

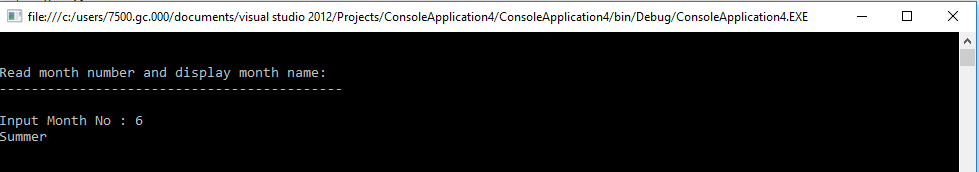


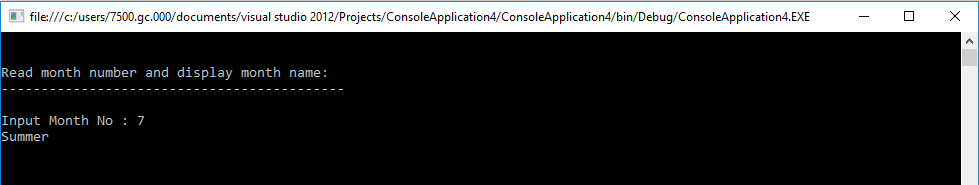


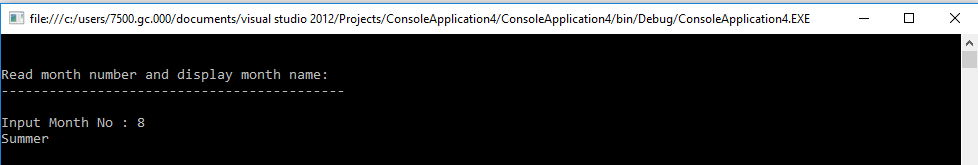


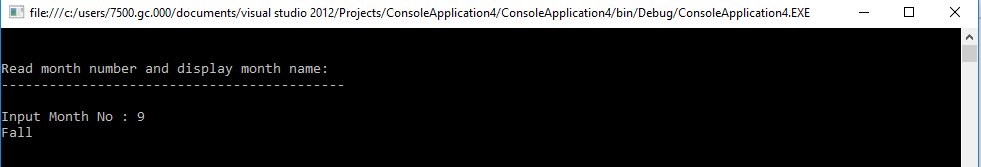


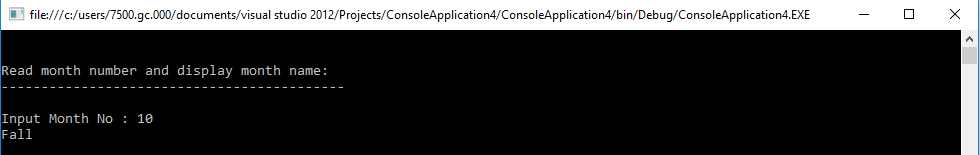


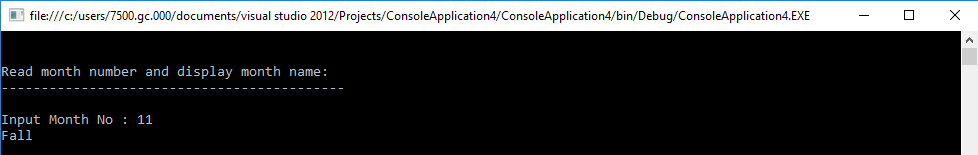


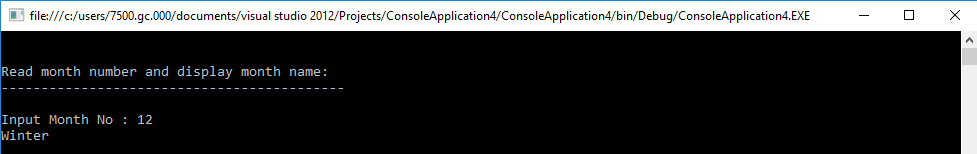












5.To determine whether a year is a leap year, follow these steps:

* 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.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace leap\_year

{

class Program

{

static void Main(string[] args)

{

int year;

Console.WriteLine("enter year:");

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

if (year % 4 == 0)

{

if (year % 100 == 0)

{

if (year % 400 == 0)

{

Console.WriteLine("The year is a leap year (it has 366 days).");

Console.ReadLine();

}

else

{

Console.WriteLine("The year is not a leap year (it has 365 days).");

Console.ReadLine();

}

}

else

{

Console.WriteLine("The year is a leap year (it has 366 days).");

Console.ReadLine();

}

}

else

{

Console.WriteLine("The year is not a leap year (it has 365 days).");

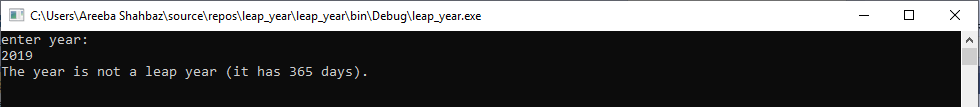
Console.ReadLine();

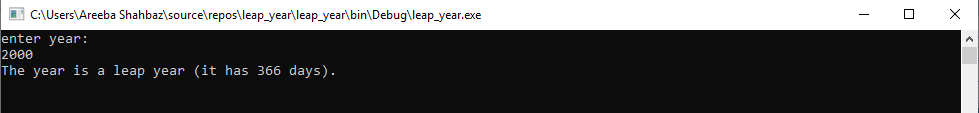
}

}

}

}





* 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.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication6

{

class Program

{

static void Main(string[] args)

{

int num1, num2, result;

char op;

Console.WriteLine("enter first number");

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

Console.WriteLine("enter op");

op = Convert.ToChar(Console.ReadLine());

Console.WriteLine("enter second number");

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

switch (op)

{

case '+':

result = num1 + num2;

Console.WriteLine("The result is "+result);

Console.ReadLine();

break;

case '-':

result = num1 - num2;

Console.WriteLine("The result is " + result);

Console.ReadLine();

break;

case '\*':

result = num1 \* num2;

Console.WriteLine("The result is " + result);

Console.ReadLine();

break;

case '/':

result = num1 / num2;

Console.WriteLine("The result is " + result);

Console.ReadLine();

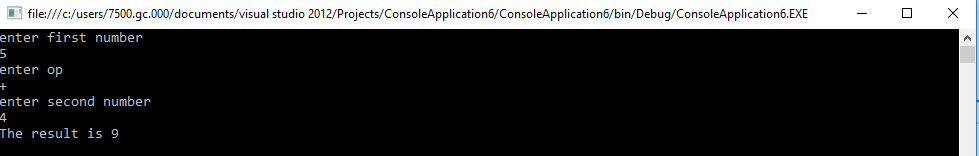
break;

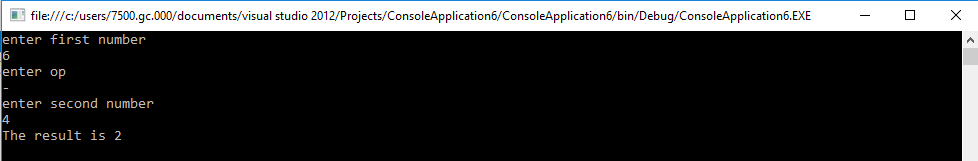
}

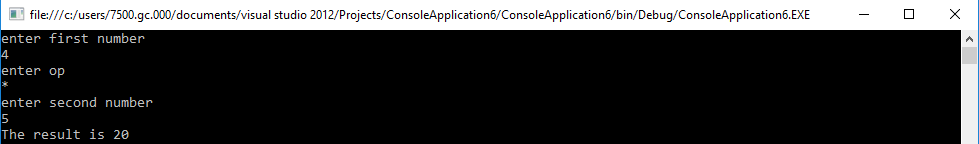
}

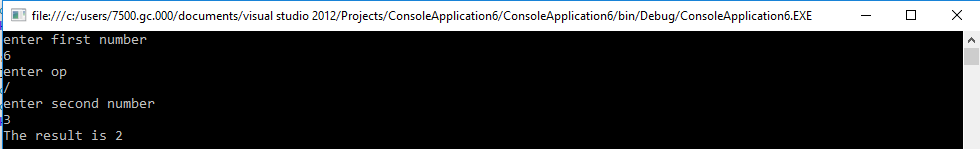
}

}









7.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 marksheet

{

class Program

{

static void Main(string[] args)

{

int marks\_of\_maths, marks\_of\_English, marks\_of\_Computer, marks\_of\_Science, marks\_of\_Social;

int percentage;

Console.WriteLine("Mark sheet");

Console.WriteLine("Maths");

marks\_of\_maths = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("English");

marks\_of\_English = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Computer");

marks\_of\_Computer = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Science");

marks\_of\_Science = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Social Studies");

marks\_of\_Social = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Percentage:");

percentage = ((marks\_of\_maths + marks\_of\_English + marks\_of\_Computer + marks\_of\_Science + marks\_of\_Social) \* 100) / 500;

if (percentage == 60)

{

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

Console.ReadLine();

}

else if (percentage <= 77)

{

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

Console.ReadLine();

}

else if (percentage <= 80)

{

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

Console.ReadLine();

}

else if (percentage <= 90)

{

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

Console.ReadLine();

}

else

{

Console.WriteLine("fail");

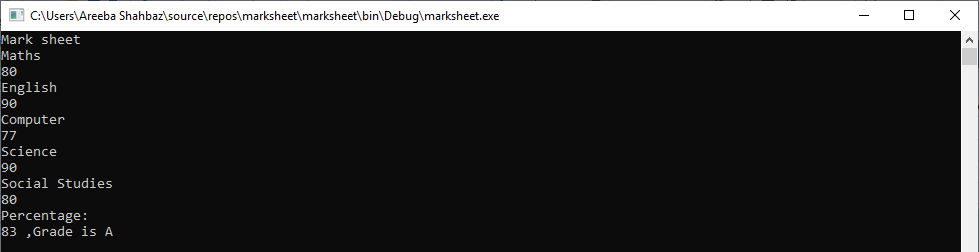
Console.ReadLine();

}

}

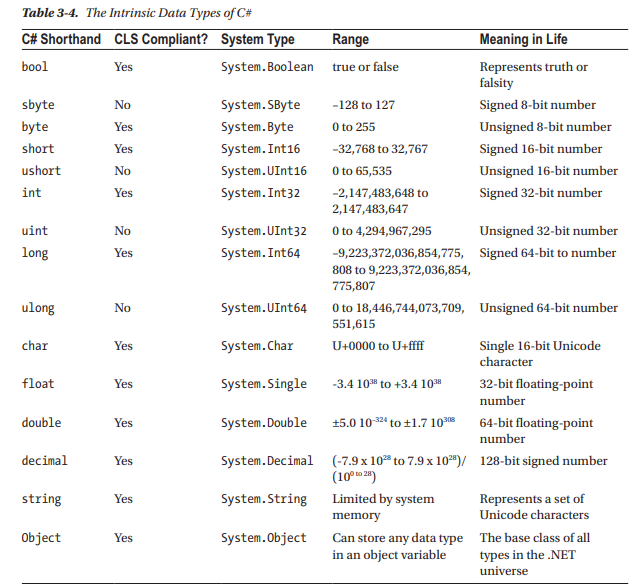
}

}

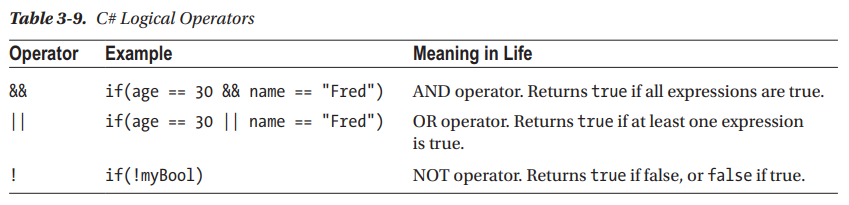


**Lab 2: To study and understand how to write programs in C# using loops, arrays and other constructs**

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.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace frequency

{

class Program

{

static int frequency(int[] a,

int n, int x)

{

int count = 0;

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

if (a[i] == x)

count++;

return count;

}

// Driver program

static public void Main()

{

int[] a = { 0, 5, 5, 5, 4 };

int x = 5;

int n = a.Length;

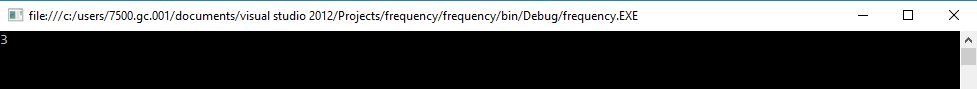
Console.Write(frequency(a, n, x));

Console.Read();

}

}

}



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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace max\_and\_min

{

class Program

{

static void Main(string[] args)

{

int[] arr1= new int[100];

int i, mx, mn, n;

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

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

n = 5;

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());

}

mx = arr1[0];

mn = arr1[0];

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

{

if(arr1[i]>mx)

{

mx = arr1[i];

}

if(arr1[i]<mn)

{

mn = arr1[i];

}

}

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

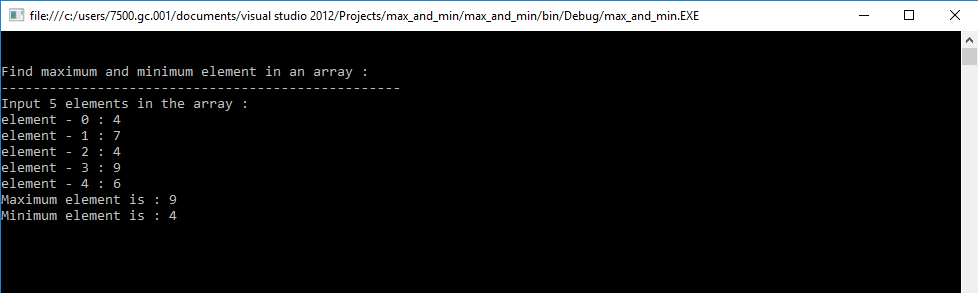
Console.Write("Minimum element is : {0}\n\n", mn);

Console.Read();

}

}

}



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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Separate\_odd\_and\_even

{

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]);

}

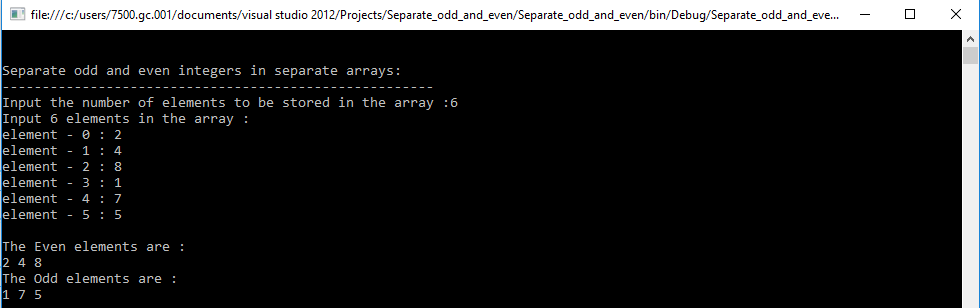
Console.Write("\n\n");

Console.Read();

}

}

}



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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace length\_of\_string

{

class Program

{

static void Main(string[] args)

{

String str;

int length = 0;

Console.WriteLine("Input your String");

str = Console.ReadLine();

foreach (char ch in str)

{

length += 1;

}

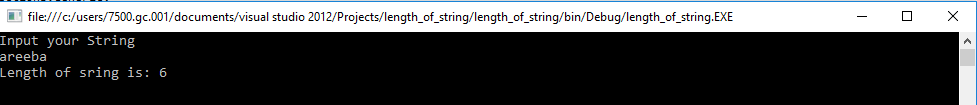
Console.Write("Length of sring is: {0}\n",length);

Console.ReadLine();

}

}

}



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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Count\_words

{

class Program

{

static int OUT = 0;

static int IN = 1;

// returns number of words in str

static int countWords(String str)

{

int state = OUT;

int wc = 0; // word count

int i = 0;

// Scan all characters one

// by one

while (i < str.Length)

{

// If next character is a separator,

// set the state as OUT

if (str[i] == ' ' || str[i] == '\n' ||

str[i] == '\t')

state = OUT;

// If next character is not a word

// separator and state is OUT, then

// set the state as IN and increment

// word count

else if (state == OUT)

{

state = IN;

++wc;

}

// Move to next character

++i;

}

return wc;

}

// Driver program to test above functions

public static void Main()

{

String str = "One two three\n four\tfive ";

Console.WriteLine("No of words : "

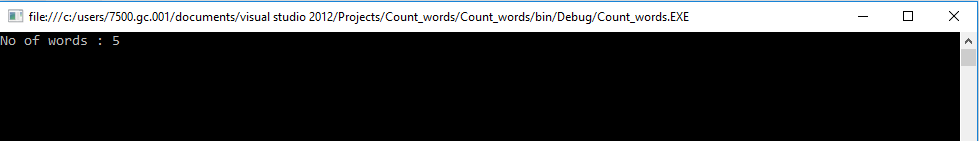
+ countWords(str));

Console.ReadLine();

}

}

}



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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Fabnocci

{

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());

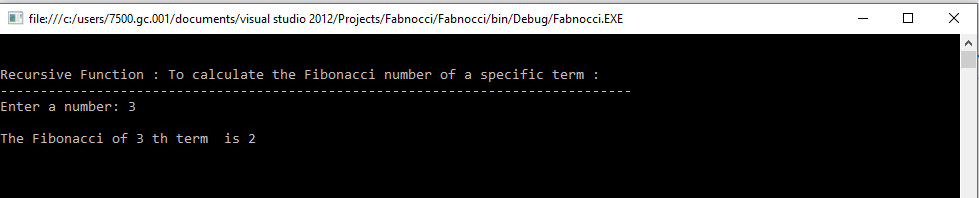
Console.WriteLine("\nThe Fibonacci of {0} th term is {1} \n", num, Fib(num));

Console.ReadLine();

}

}

}



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



namespace ConsoleApplication18

{

class circle

{

private float r;

public float radius

{

get

{

return r;

}

set

{

if (value < 0)

throw new ArgumentException("Invalid radius");

}

}

}

}

* + - 1. Define a one argument constructor to initialize the radius.

class circle

{

public circle(float radius)

{

this.radius = radius;

}

public circle()

: this(0)

{

}

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

class Program

{

static void Main(string[] args)

{

circle c = new circle();

c.radius = 9;

Console.Write(c.radius);

}

}

}

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

public float GetArea(){

return 3.14f\* radius \*radius;

}

}

}

Main class

Console.WriteLine(c.GetArea());

}

* + - 1. Create a Test class.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication18

{

class test

{

}

}

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

static void Main(string[] args)

{

circle c = new circle();

c.radius = 9;

Console.Write(c.radius);

}

}

}

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

circle c1 = new circle();

c1.r = 6;

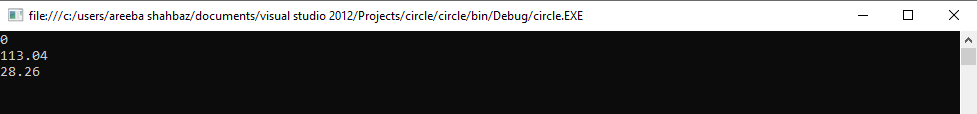
Console.WriteLine(c1.getarea());

circle c2 = new circle();

c2.r = 3;

Console.WriteLine(c2.getarea());

Console.ReadLine();



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.

Solution For A To E

class Animal

{

protected boolean isMammal;

protected boolean isCarnivorous;

public Animal(boolean isMammal, boolean isCarnivorous) {

this.isMammal = isMammal;

this.isCarnivorous = isCarnivorous;

}

public boolean getIsMammal() {

return this.isMammal;

}

public boolean getIsCarnivorous() {

return this.isCarnivorous;

}

abstract public String getGreeting();

public void printAnimal(String name) {

Console.Write("A " + name + " says '" + this.getGreeting() + "', is " + (this.getIsCarnivorous() ? "" : "not ")

+ "carnivorous, and is " + (this.getIsMammal() ? "" : "not ") + "a mammal.");

}

}

class goat : Animal{

public goat() {

super(true, true);

}

public String getGreeting() {

return "mee";

}

}

class Cat : Animal{

public Cat() {

super(true, false);

}

public String getGreeting() {

return "meewoo";

}

}

class fish : Animal{

public fish() {

super(false, false);

}

public String getGreeting() {

return "zzzzzz";

}

}

public class AnimalInheritance{

public static void main(String[] args) {

Animal cat = new cat();

cat.printAnimal("cat");

Animal goat = new goat();

goat.printAnimal("g");

Animal fish = new fish();

fish.printAnimal("fish");

}

}

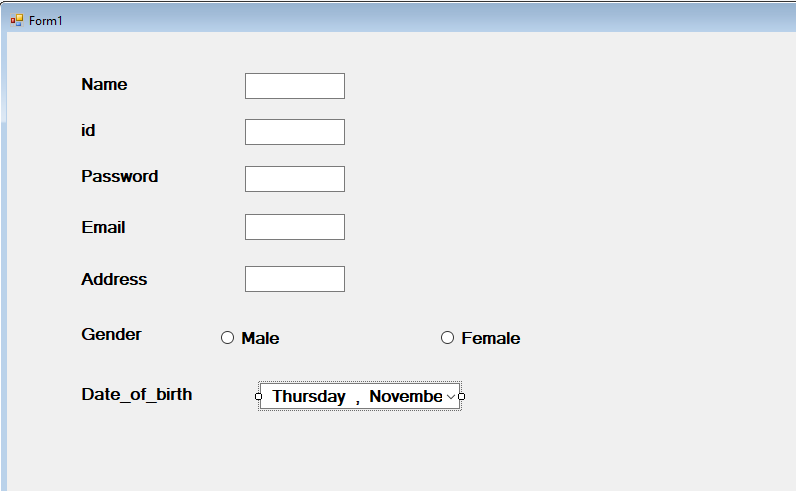
**Lab 4: To study and implement Windows Forms application in C#**

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.

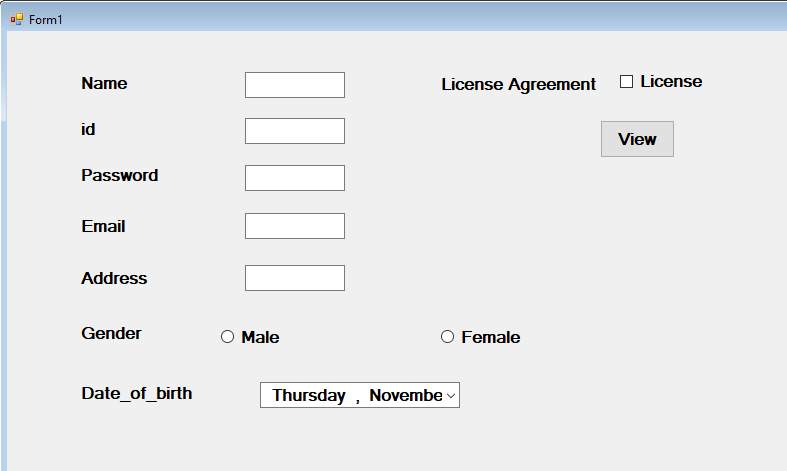
We will look at some of the most common controls used in C# and use them to develop a basic application.

**Tasks:**

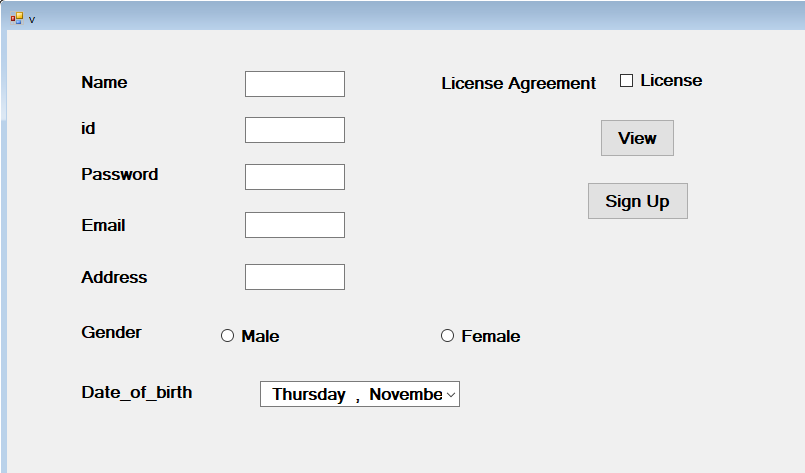
1. Design a basic Sign Up form with fields name, id, password, email, address, gender and date of birth.



1. Add a checkbox to accept license agreement, and a view button to view the license agreement.



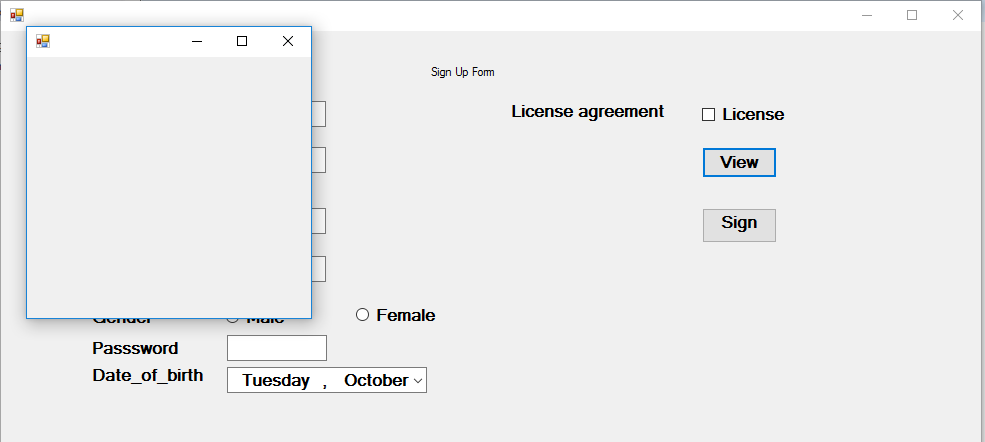
1. Add a Sign Up button.



1. 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

1. using System;
2. using System.Collections.Generic;
3. using System.ComponentModel;
4. using System.Data;
5. using System.Drawing;
6. using System.Linq;
7. using System.Text;
8. using System.Threading.Tasks;
9. using System.Windows.Forms;
10. namespace WindowsFormsApplication1
11. {
12. public partial class Signup\_form : Form
13. {
14. public Signup\_form()
15. {
16. InitializeComponent();
17. }
18. private void label1\_Click(object sender, EventArgs e)
19. {
20. }
21. private void Signupbutton\_Click(object sender, EventArgs e)
22. {
23. string name = Namebox.Text;
24. string id = idbox.Text;
25. string email = this.Emailbox.Text;
26. string age = this.dateTimePicker1.Text;
27. string pass = this.Passwordbox.Text;
28. string dob = this.dateTimePicker1.Text;
29. string g = "Male";
30. if (this.MaleButton.Checked)
31. {
32. g = "Female";
33. }
34. bool Licen = this.License.Checked;
35. if (name == "" || email == "" || dob == "" || pass == "" || id == "" || g == "" || pass == "" || Licen == false)
36. {
37. MessageBox.Show("Please fill all the fields");
38. return;
39. }
40. if (!email.Contains("@") ||
41. !email.Contains("."))
42. {
43. MessageBox.Show("Please provide valid email address");
44. return;
45. }
46. }
47. private void Viewbutton\_Click(object sender, EventArgs e)
48. {
49. Form f = new Form();
50. f.Show();
51. }
52. }
53. }
54. When the user clicks on view license agreement button, a new form is to be displayed with the license agreement.
55. private void Viewbutton\_Click(object sender, EventArgs e)
56. {
57. Form f = new Form();
58. f.Show();
59. }

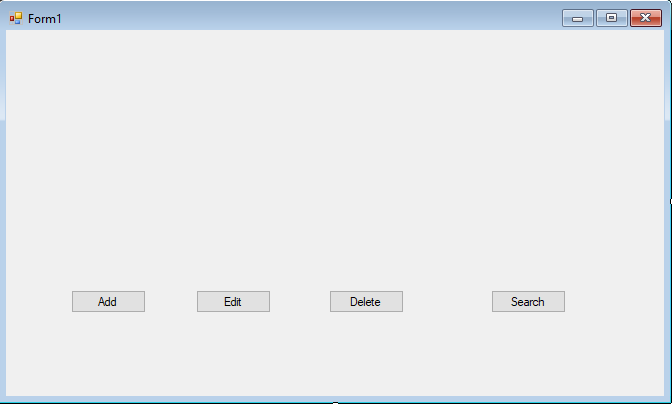


**Lab 5: To study and implement Collections in C#**

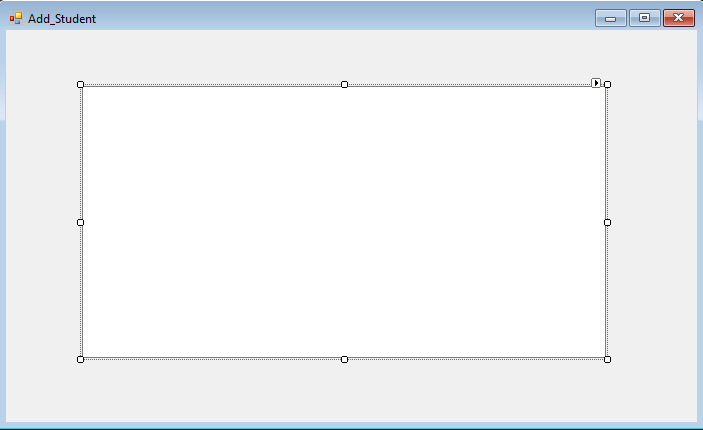
In todays, lab we will implement the collections in C#. A collection is an abstract data type for grouping together multiple values. It's therefore sometime known as container. A collection is just a grouping of some objects with the same type.

**Lab Tasks:**

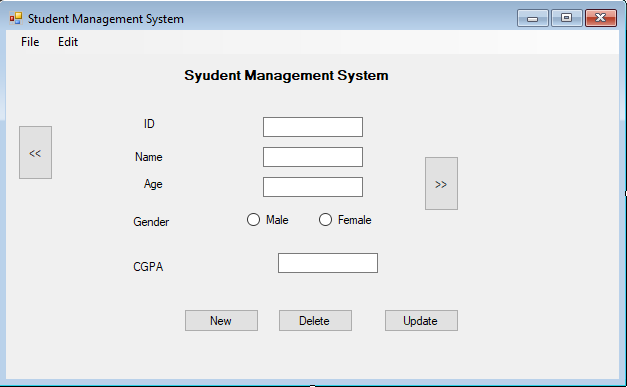
1. Create a WinForm in C# with following buttons: Add a Student, Edit a Student, Delete a Student, Search a Student.



1. Now, create a class Student with the following properties: name, age, gender, cgpa
2. using System;
3. using System.Collections.Generic;
4. using System.Linq;
5. using System.Text;
6. using System.Threading.Tasks;
7. namespace Student\_Management\_System
8. {
9. class Student
10. {
11. public string name { get; set; }
12. public int age { get; set; }
13. public bool gender { get; set; }
14. public decimal cgpa { get; set; }
15. public Student(string name, int age, bool gender, decimal cgpa)
16. {
17. this.name = name;
18. this.age = age;
19. this.gender = gender;
20. this.cgpa = cgpa;
21. }
22. }
23. }
24. 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.



1. Implement the Edit button such that user can modify the students record. The changes are reflected in the collection.
2. Implement the Delete and Search options.



//Student class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WindowsFormsApplication3

{

[Serializable()]

class Student

{

public string id { get; set; }

public string name { get; set; }

public int age { get; set; }

public bool gender { get; set; }

public decimal cgpa { get; set; }

public Student(string id, string name, int age, bool gender, decimal cgpa)

{

this.id = id;

this.name = name;

this.age = age;

this.gender = gender;

this.cgpa = cgpa;

}

}

}

//Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApplication3

{

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());

}

}

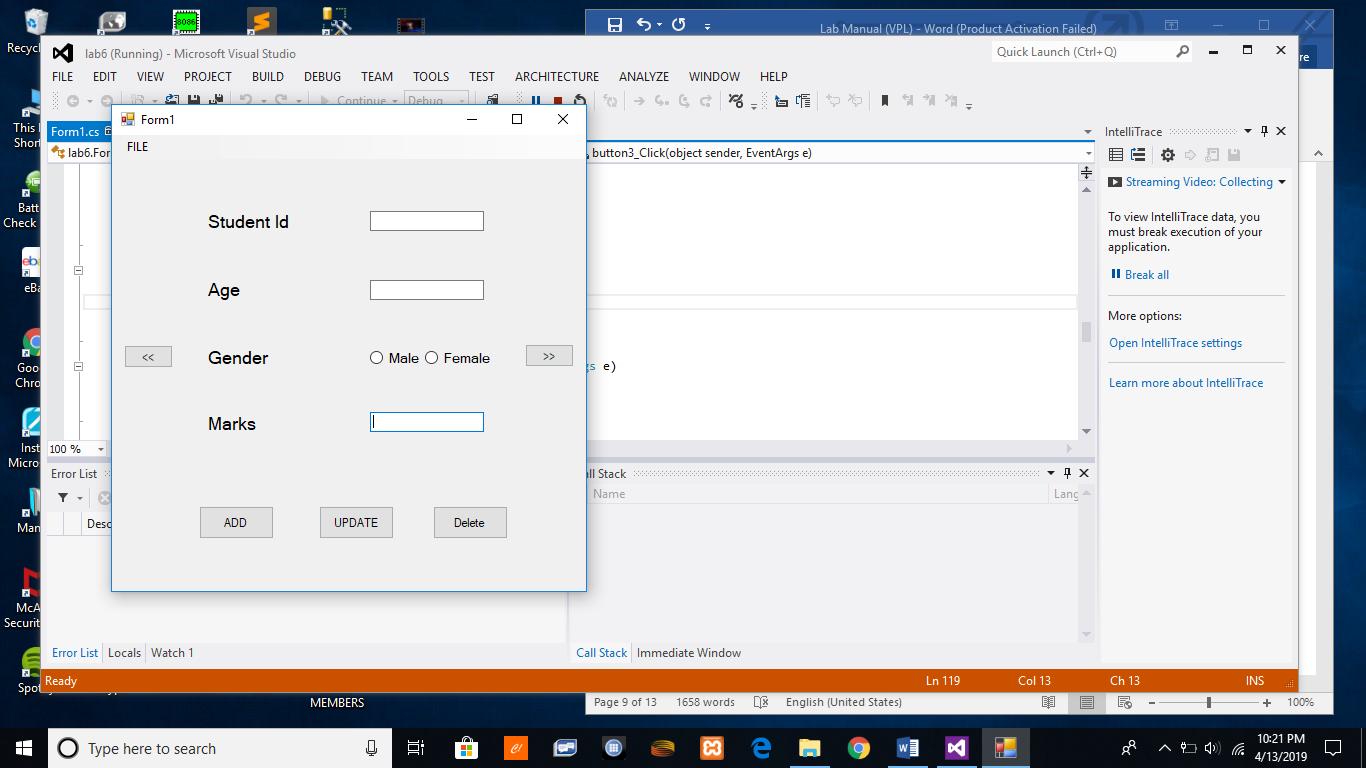
}

**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. using System;
5. using System.Collections.Generic;
6. using System.ComponentModel;
7. using System.Data;
8. using System.Drawing;
9. using System.Linq;
10. using System.Text;
11. using System.Threading.Tasks;
12. using System.Windows.Forms;
13. using System.IO;
14. namespace lab6
15. {
16. public partial class Form1 : Form
17. {
18. List<Student> students = new List<Student>();
19. int current =-1;
20. public Form1()
21. {
22. InitializeComponent();
23. }
24. private void Form1\_Load(object sender, EventArgs e)
25. {
26. }
27. private void button1\_Click(object sender, EventArgs e)
28. {
29. Student std = new Student();
30. std.id = textBox1.Text;
31. std.age = textBox2.Text;
32. if (Male.Checked)
33. {
34. std.gender = "male";
35. }
36. else {
37. std.gender = "female";
38. }
39. std.marks = textBox3.Text;
40. students.Add(std);
41. current = students.Count - 1;
42. MessageBox.Show("Student added successfully");
44. }
45. private void button5\_Click(object sender, EventArgs e)
46. {
48. if (current >= students.Count-1)
49. {
50. return;
52. }
53. else {
54. current++;
55. Student s = students[current];
56. textBox1.Text = s.id;
57. textBox2.Text = s.age;
58. textBox3.Text = s.marks;
59. if (s.gender == "male")
60. {
61. Male.Checked = true;
62. }
63. else
64. {
65. Female.Checked = true; } }
66. }
67. private void button4\_Click(object sender, EventArgs e)
68. {
69. if (current <= 0)
70. {
71. return;
72. }
73. else {
74. current--;
75. Student s = students[current];
76. textBox1.Text = s.id;
77. textBox2.Text = s.age;
78. textBox3.Text = s.marks;
79. if (s.gender == "male")
80. {
81. Male.Checked = true;
82. }
83. else
84. {
85. Female.Checked = true;
86. }
87. }
88. }
89. private void button2\_Click(object sender, EventArgs e)
90. {
91. Student s = students[current];
92. s.id = textBox1.Text;
93. s.age = textBox2.Text;
94. s.marks = textBox3.Text;
95. if (Male.Checked)
96. {
97. s.gender = "Male";
98. }
99. else {
100. s.gender = "Female";
101. }
102. }
103. private void button3\_Click(object sender, EventArgs e)
104. {
105. students.RemoveAt(current);
106. current--;
107. }
108. private void oPENToolStripMenuItem\_Click(object sender, EventArgs e)
109. {
110. }
111. private void eXITToolStripMenuItem1\_Click(object sender, EventArgs e)
112. {
113. this.Close();
114. }
115. private void eXITToolStripMenuItem\_Click(object sender, EventArgs e)
116. {
117. if (saveFileDialog1.ShowDialog() == DialogResult.OK)
118. {
119. StreamWriter sw = new StreamWriter(saveFileDialog1.FileName,true);
120. foreach (Student s in students)
121. {
122. sw.WriteLine(s.id + " " + s.age + " " + s.marks+ " "+s.gender);
123. }
124. sw.Close();
125. MessageBox.Show("File Saved Successfully");
126. }
127. }
128. private void saveFileDialog1\_FileOk(object sender, CancelEventArgs e)
129. {
130. }
131. private void sAVEToolStripMenuItem\_Click(object sender, EventArgs e)
132. {
133. if (openFileDialog1.ShowDialog() == DialogResult.OK)
134. {
135. StreamReader sr = new StreamReader(openFileDialog1.FileName,true);
137. string line = sr.ReadLine();
138. students = new List<Student>();
140. while (line != null)
141. {
142. string[] tokens = line.Split();
143. Student s = new Student();
144. s.id = tokens[0];
145. s.age = tokens[1];
146. s.marks = tokens[2];
147. s.gender = tokens[3];
148. students.Add(s);
149. line = sr.ReadLine();
150. }
151. current = 0; } } }}
152. STUDENT CLASS:
153. using System;
154. using System.Collections.Generic;
155. using System.Linq;
156. using System.Text;
157. using System.Threading.Tasks;
158. namespace lab6
159. {
160. class Student
161. {
162. public string id;
163. public string age;
164. public string gender;
165. public string marks;
166. }
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

<?xml version="1.0"?>

<configuration>

<background>red</background>

<foreground>blue</foreground>

<text>hello</text>

<title>my title</title>

<width>100</width>

<height>100</height>

</configuration>

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 WindowsFormsApplication8

{

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);

}

}

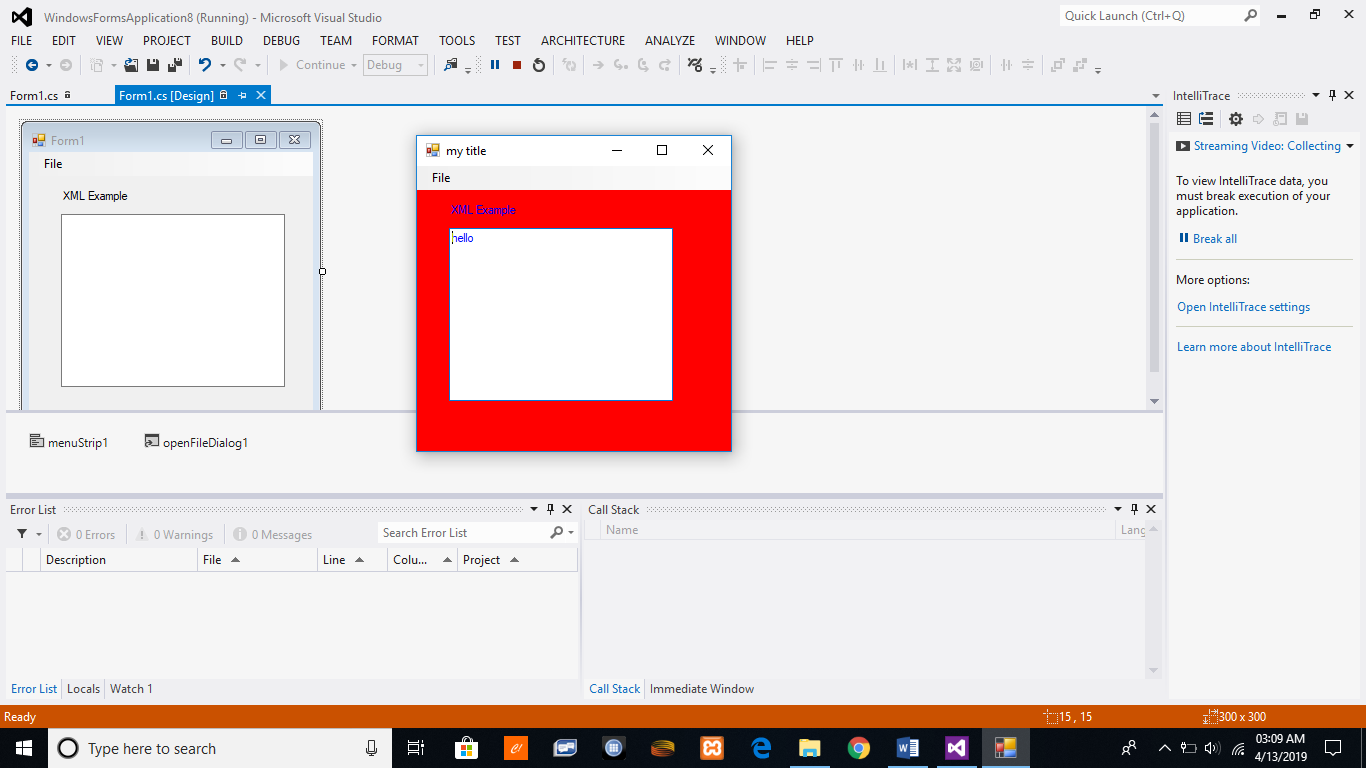
private void Form1\_Load(object sender, EventArgs e)

{

} }

}

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.

**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;}

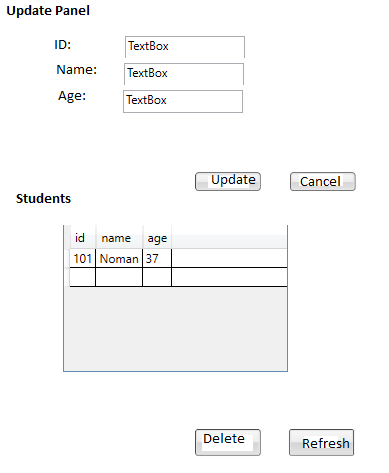
public string name {set; get;}

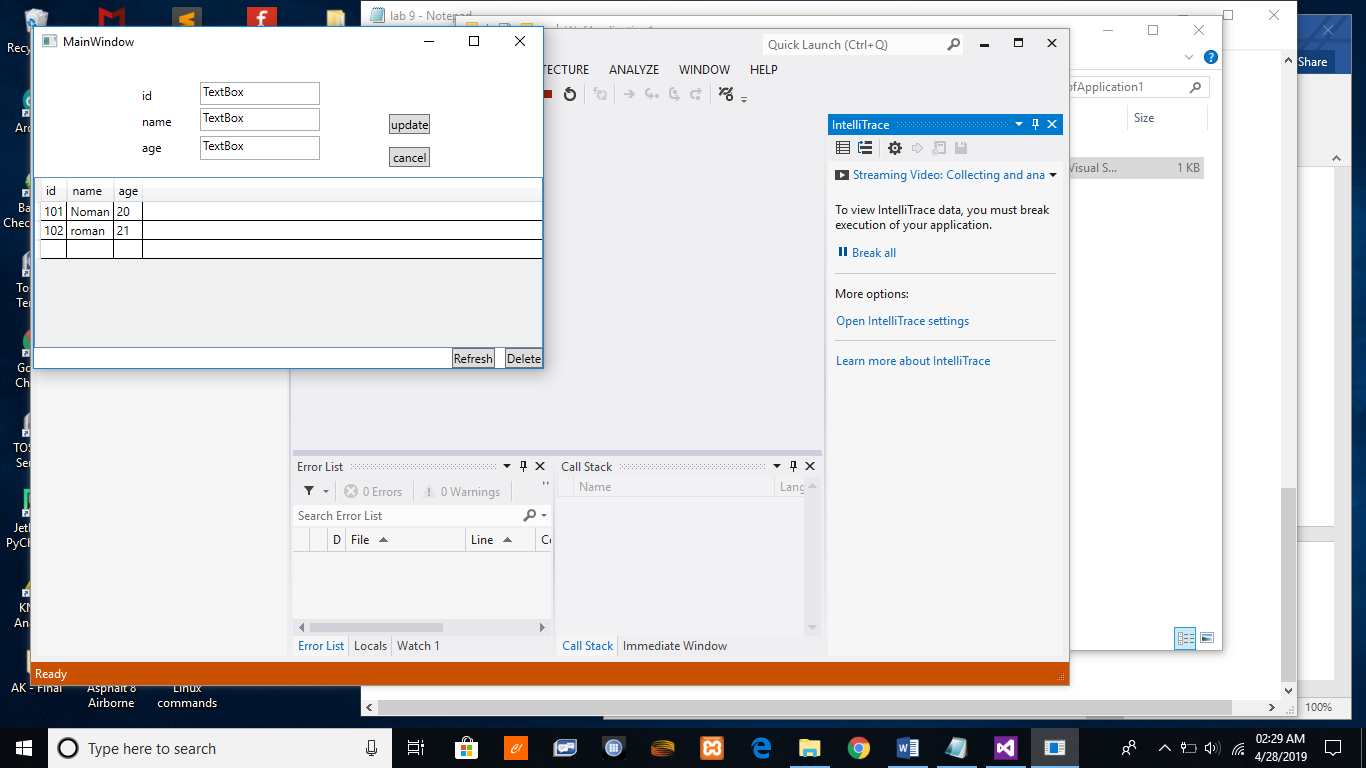
public string age {set; get;}

}

}

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:**

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.

**Code**

task 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;

}

}

}

Task 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 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;

}

}

}

Code

Output

