

Visual Programming Languages

Lab Manual

[Fall 2019]

|  |  |
| --- | --- |
| Student Name: Umme Ruman |  |
|  |  |
| Student Id: 12887 |  |
|  |  |

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

LIST OF EXPERIMENTS

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Date** | **Experiment** |  |
| **1** | 3/02/19 | To setup the environment and familiarize with C# |  |
| **2** | 10/02/19 | To study and understand how to write programs in C# using loops and arrays |  |
| **3** | 17/02/19 | To study and implement object oriented programming concepts in C# |  |
| **4** | 17/03/19 | To study and implement Windows Forms application in C# |  |
| **5** | 24/03/19 | To study and implement Collections in C# |  |
| **6** | 7/04/19 | To study and implement I/O in C# |  |
| **7** | 14/04/19 | To study and implement XML parsing in C# |  |
| **8** | 21/04/19 | To study and implement WPF and its layouts in C# |  |
| **9** | 28/04/19 | To study and implement LINQ 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.

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Umme Ruman Talib \n \_\_\_\_\_\_\_ \n Developer");

Console.WriteLine("\nEducation \ABC college

College" +

" Secondary School 2016 \nMatriculation: ABC School" );

Console.WriteLine("\nSkills \nJava, C, C#, Python, HTML" +

"CSS Languages");

Console.WriteLine("\nExperience: freshee");

Console.WriteLine("\nContact \n001122334455\urt @gmail.com" +

"\nlinkedin.com \nKarachi, Pakistan");

}

}

}

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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab12

{

class Program

{

static void Main(string[] args)

{

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

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

if (num % 2 == 0)

Console.WriteLine("Entered number is even !!!!!");

else

Console.WriteLine("Entered number is odd!!!!! ");

}

}

}

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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab13

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter any 3 numbers: ");

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

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

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

if (a >= b && a >= c)

Console.WriteLine("largest num is " + a);

else if (b >= a && b >= c)

Console.WriteLine("largest num is " + b);

else

Console.WriteLine("largest num is " + c);

if (a <= b && a <= c)

Console.WriteLine("smallest num is " + a);

else if (b <= a && b <= c)

Console.WriteLine("smallest num is " + b);

else

Console.WriteLine("smallest num is " + c);

}

}

}

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.

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab14

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter month number: ");

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

if (m == 12 || m == 1 || m == 2)

Console.WriteLine("It’s Winter Season!!!!!!");

else if (m == 3 || m == 4 || m == 5)

Console.WriteLine("It’s Spring Season!!!!!");

else if (m == 6 || m == 7 || m == 8)

Console.WriteLine("It’s Summer Season!!!!!");

else

Console.WriteLine(“It’s Autumn Season!!!!!");

}

}

}

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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab15

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the year: ");

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

if (x % 4 == 0)

{

if (x % 100 == 0)

{

if (x % 400 == 0)

{

Console.Write(x + " is a leap year!");

}

else

Console.WriteLine(x+" is not a leap year, it has 365 days!");

}

else

Console.WriteLine(x+" is a leap year it has 366 days!");

}

else

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

}

}

}

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.

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab16

{

class Program

{

static void Main(string[] args)

{

int result;

Console.WriteLine("Enter any two numbers: ");

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

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

Console.WriteLine("Enter any operator: ");

String ch = Console.ReadLine();

switch(ch)

{

case "+":

result = num1 + num2;

Console.WriteLine("sum of " + num1 + " and " + num2 + " is: " + result);

break;

case "-":

result = num1 - num2;

Console.WriteLine("subtraction of " + num1 + " and " + num2 + " is: " + result);

break;

case "/":

result = num1 / num2;

Console.WriteLine("division of " + num1 + " and " + num2 + " is: " + result);

break;

case "\*":

result = num1 \* num2;

Console.WriteLine("multiplication of " + num1 + " and " + num2 + " is: " + result);

break;

}

}

}

}

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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab17

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter English Marks: ");

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

Console.Write("Enter Urdu Marks: ");

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

Console.Write("Enter Maths Marks: ");

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

Console.Write("Enter Physics Marks: ");

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

Console.Write("Enter Islamiat Marks: ");

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

int total=eng + ur + math + phy + isl;

Console.WriteLine("\nObtained marks= " + total);

float per = total \* 100 / 500;

Console.WriteLine("Percentage= " + per);

if (per >= 80)

Console.WriteLine("Grade is A");

else if (per >= 70)

Console.WriteLine("Grade is B");

else if (per >= 60)

Console.WriteLine("Grade is C");

else if (per >= 50)

Console.WriteLine("Grade is D");

else if (per >= 40)

Console.WriteLine("Grade is E");

else

Console.WriteLine("You are Fail");

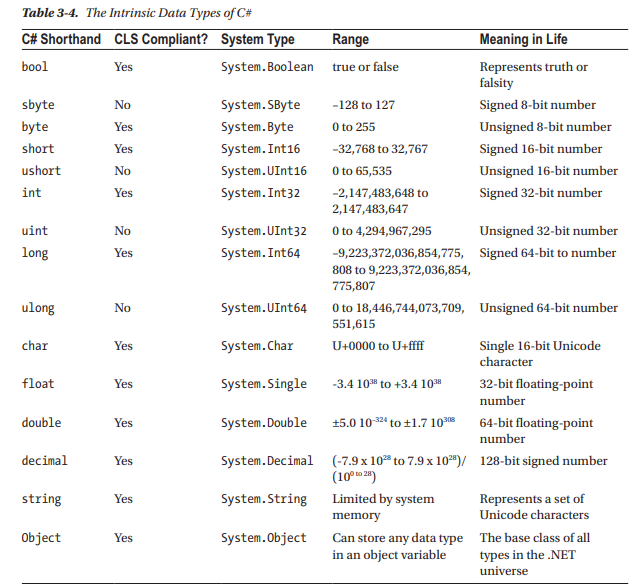
}

}

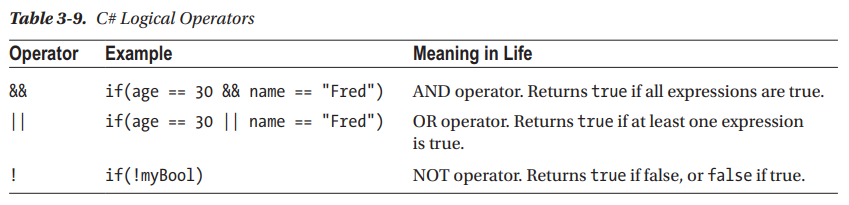
}

**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.

**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)

{

int[] x = { 3, 5, 3, 6, 3, 4, 5, 10, 8, 7 };

foreach (int i in x.Distinct().ToArray())

{

int c = 0;

foreach (int j in x)

{

if (i == j)

{

c++;

}

}

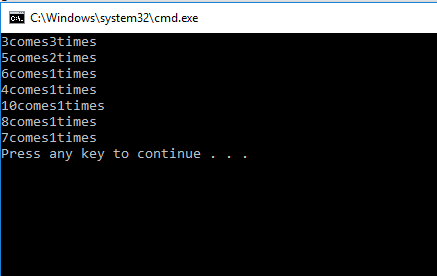
Console.WriteLine(i + "comes" + c + "times");

}

}

}

}



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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication11

{

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;

}

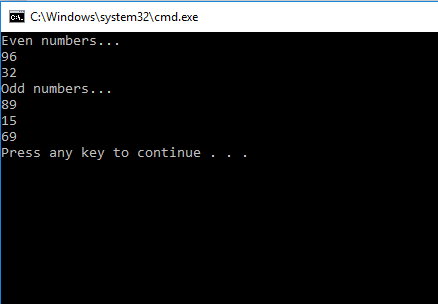
Console.Write("Length of the string is : {0}\n\n", length);

Console.ReadLine();

}

}

}



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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication10

{

class Program

{

static void Main(string[] args)

{

int[] arr1 = new int[] {

89,

15,

69,

96,

32

};

int[] arr2 = new int[5];

int[] arr3 = new int[5];

int i, j = 0, k = 0;

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

{

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

{

arr2[j] = arr1[i];

j++;

}

else

{

arr3[k] = arr1[i];

k++;

}

}

Console.WriteLine("Even numbers...");

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

{

Console.WriteLine(arr2[i]);

}

Console.WriteLine("Odd numbers...");

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

{

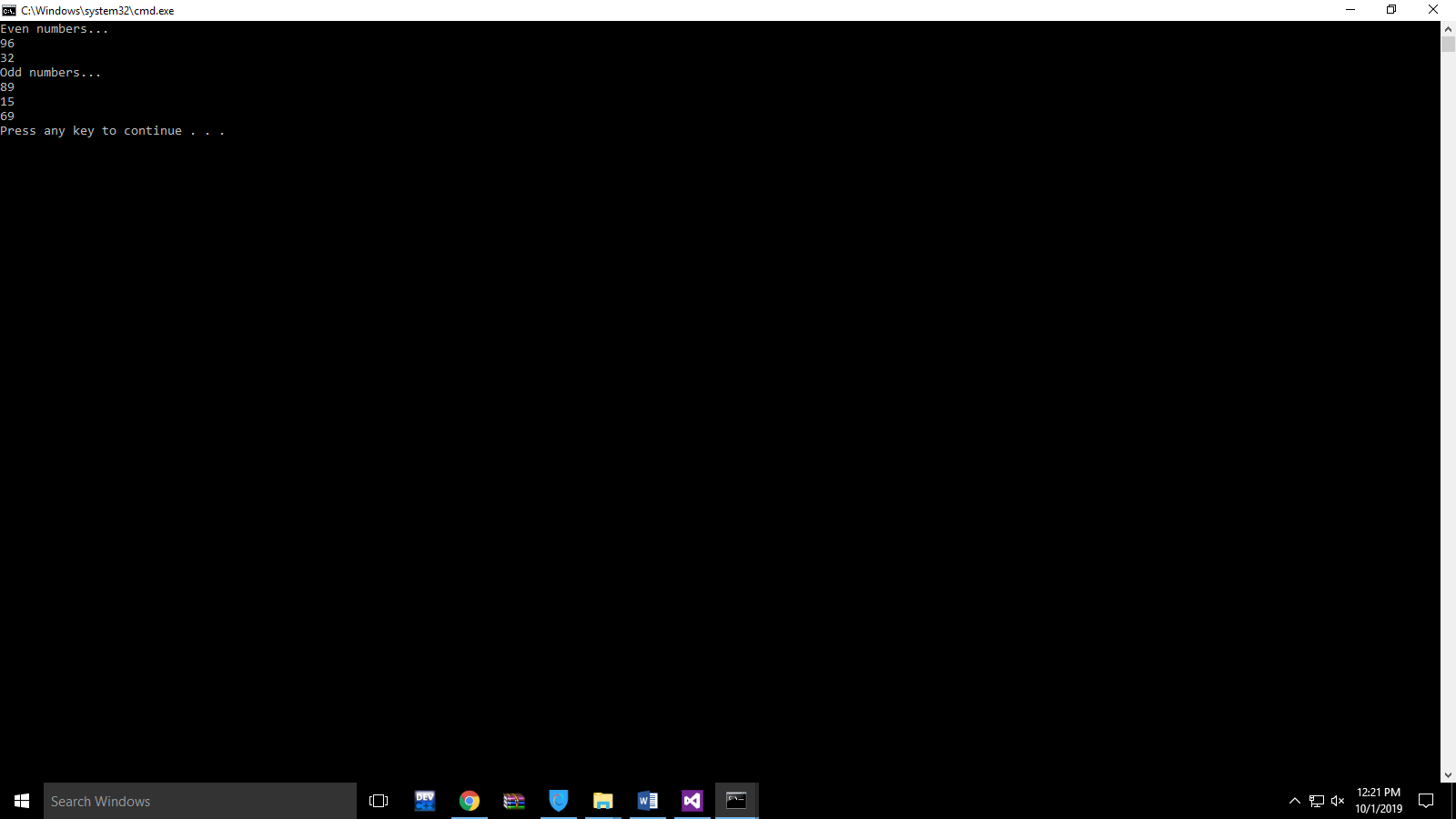
Console.WriteLine(arr3[i]);

}

}

}

}



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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication11

{

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;

}

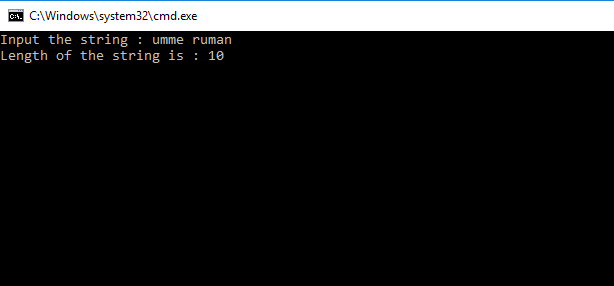
Console.Write("Length of the string is : {0}\n\n", length);

Console.ReadLine();

}

}

}



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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication12

{

class Program

{

static void Main(string[] args)

{

int a = 0 , myWord = 1;

string str = "Umme Ruman Talib";

while (a <= str.Length - 1) {

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

myWord++;

}

a++;

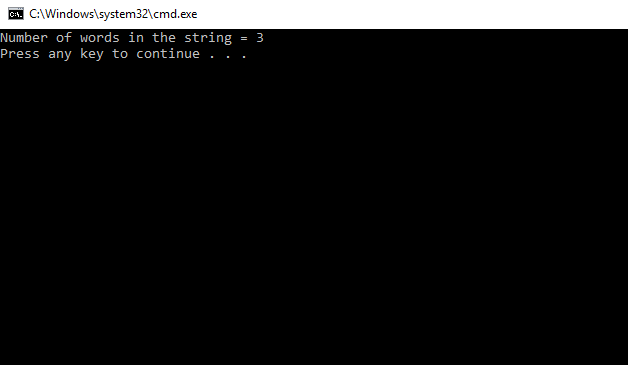
}

Console.Write("Number of words in the string = {0}\n", myWord);

}

}

}



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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication13

{

class Program

{

static void Main(string[] args)

{

int n, i = 0, c;

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

n = Convert.ToInt16(Console.ReadLine());

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

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

{

int result = FibonacciFunction(i);

Console.Write(result + " " );

i++;

}

Console.WriteLine();

}

public static int FibonacciFunction(int n)

{

if (n == 0)

{

return 0;

}

else if (n == 1)

{

return 1;

}

else

{

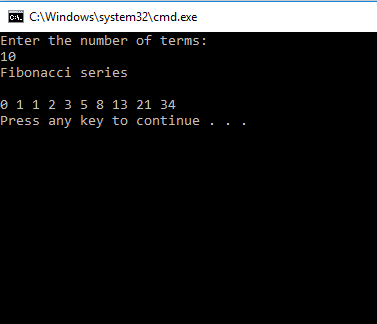
return (FibonacciFunction(n - 1) + FibonacciFunction(n - 2));

}

}

}

}



**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.
       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.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication25

{

class Circle

{

public Circle(float radius)

{

this.radius = radius;

}

public Circle()

: this(0)

{

}

private float r;

public float radius

{

get

{

return r;

}

set

{

if (value < 0)

{

throw new ArgumentException("Invalid radius");

}

else

{

r = value;

}

}

}

public float GetArea()

{

float area = 3.142f \* r \* r;

return area;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication25

{

class test

{

static void Main(string[] args)

{

Circle c = new Circle();

c.radius = 9;

Console.WriteLine(c.radius);

Console.WriteLine(c.GetArea());

Circle c1 = new Circle();

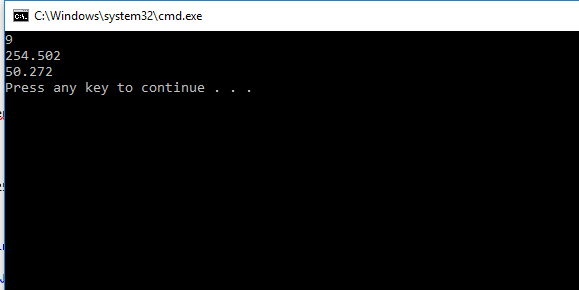
c1.radius = 4;

Console.WriteLine(c1.GetArea());

}

}

}



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

**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.

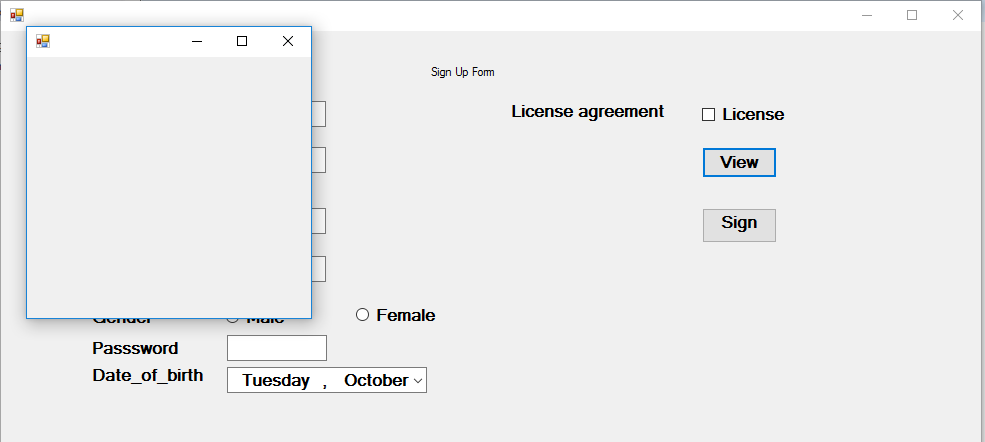
We will look at some of the most common controls used in C# and use them to develop a basic application.

**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.

**Code:**

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. private void Viewbutton\_Click(object sender, EventArgs e)
55. {
56. Form f = new Form();
57. f.Show();
58. }



**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.

**Code:**

**Form Code:**

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);

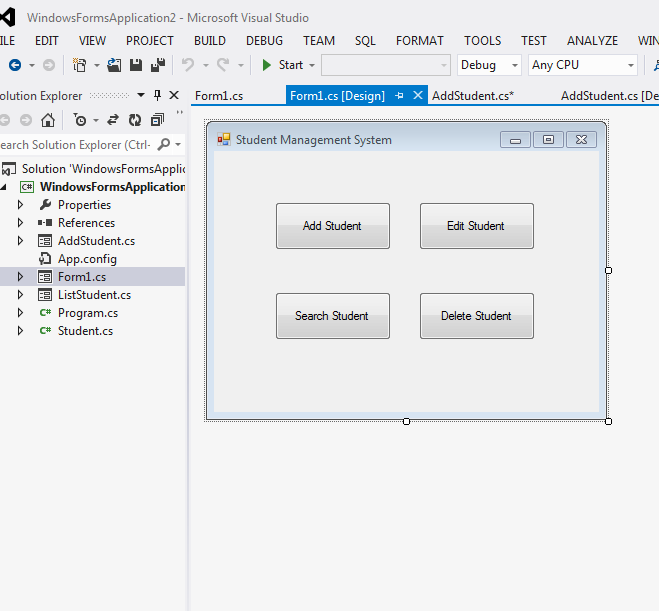
ls.select\_delete();

ls.Show();

}

}

}

****

**Add Student class:**

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();

}

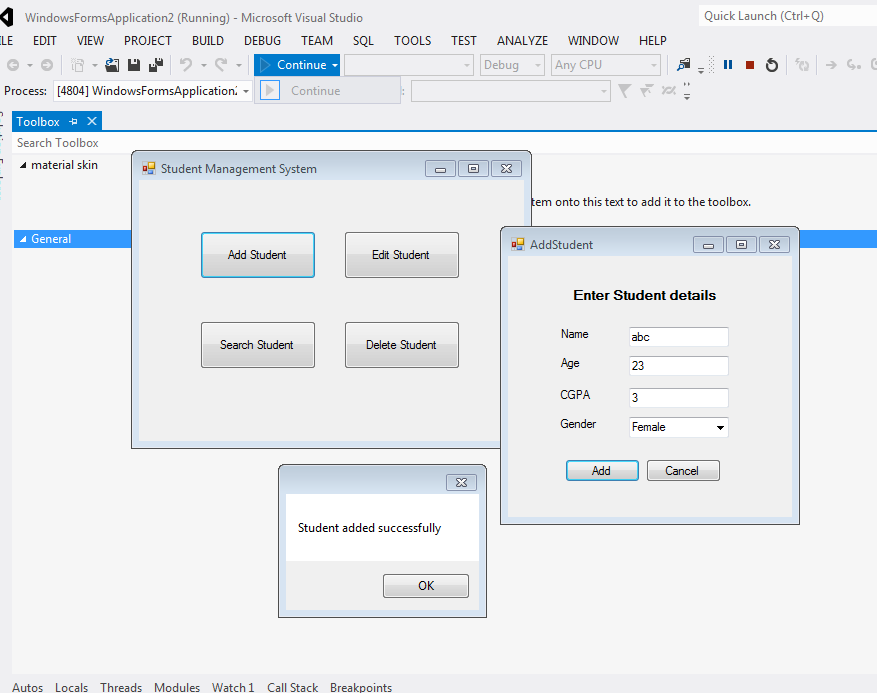
private void AddStudent\_Load(object sender, EventArgs e)

{

}

}

}

****

**List student class:**

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);

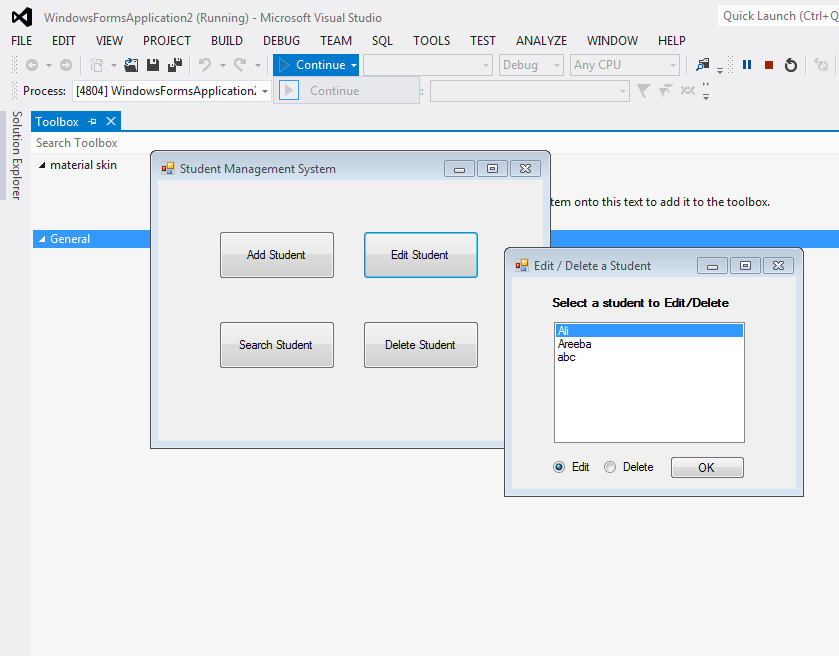
add.Show();

}

}

}

}

****

**Student class:**

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;

}

}

}

**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.

**Code:**

**Form 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;

namespace WindowsFormsApplication3

{

public partial class Form1 : Form

{

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

private int current = 0;

public Form1()

{

InitializeComponent();

}

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

{

students.Add(new Student("0","Subhan",20,true,3.5m));

students.Add(new Student("1", "Sehar", 18, false, 3.9m));

this.refresh\_form();

}

private void refresh\_form()

{

Student s;

if (students.Count == 0)

{

s = new Student("", "", 18, true, 0.0m);

}

else

{

s = students[current];

}

this.sid.Text = s.id;

this.sname.Text = s.name;

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

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

this.smale.Checked = s.gender;

this.sfemale.Checked = !s.gender;

}

//next

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

{

if (current == students.Count - 1)

{

return;

}

current++;

this.refresh\_form();

}

//previous

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

{

if (current == 0)

{

return;

}

this.current--;

this.refresh\_form();

}

//update

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

{

Student s = students[current];

s.id= this.sid.Text;

s.name = this.sname.Text;

s.gender = this.smale.Checked;

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

s.cgpa = decimal.Parse(this.scgpa.Text);

MessageBox.Show("Record has been updated");

}

//delete

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

{

if (current == students.Count - 1)

{

this.students.RemoveAt(current);

current = 0;

}

else

{

this.students.RemoveAt(current);

}

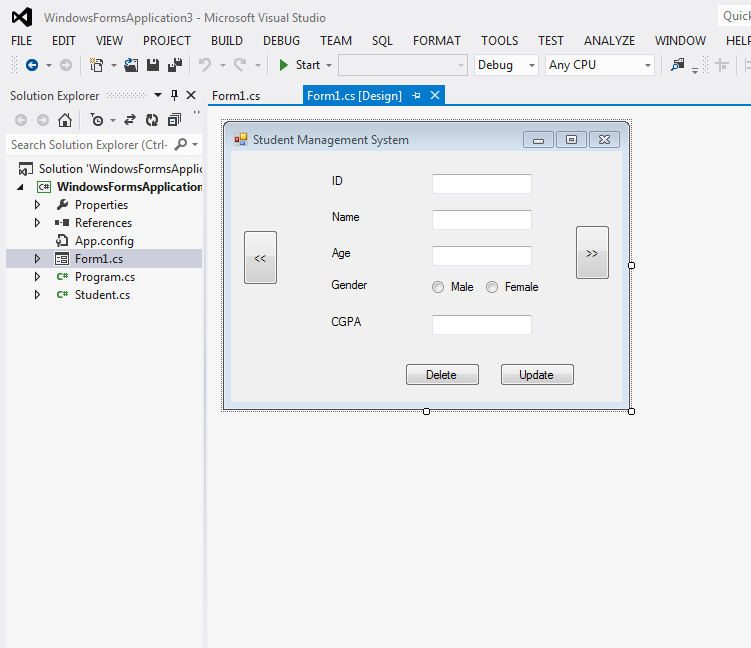
MessageBox.Show("The record has been deleted");

this.refresh\_form();

}

}

}

****

**Student class:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WindowsFormsApplication3

{

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;

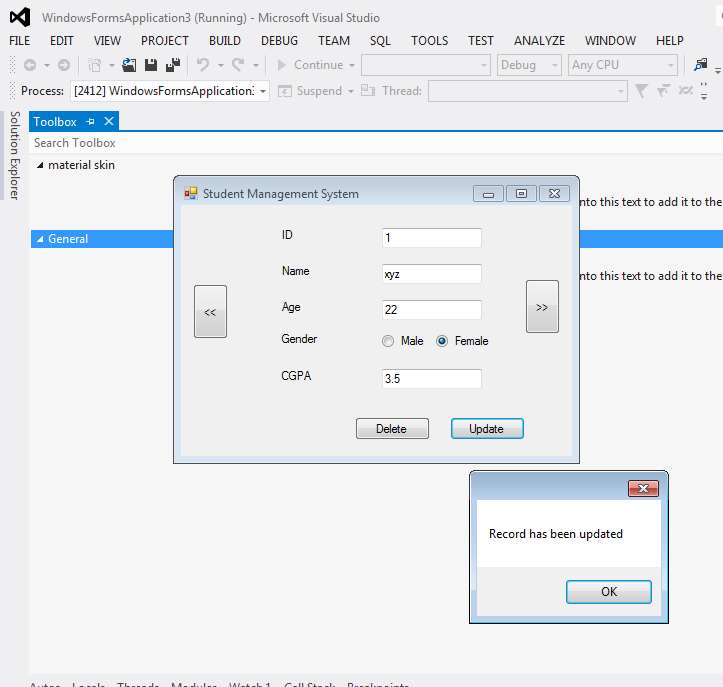
this.gender = gender;

this.cgpa = cgpa;

}

}

}

****

**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);

}

}

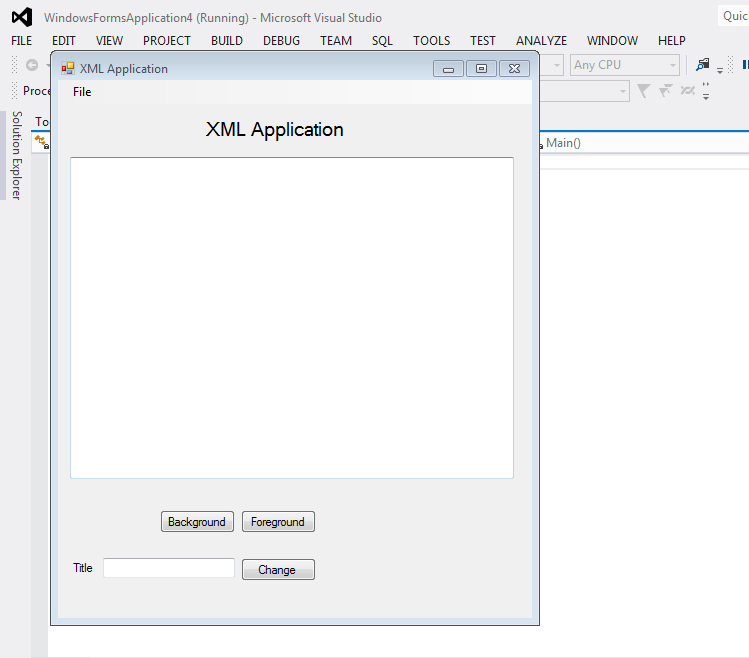
private void menuStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

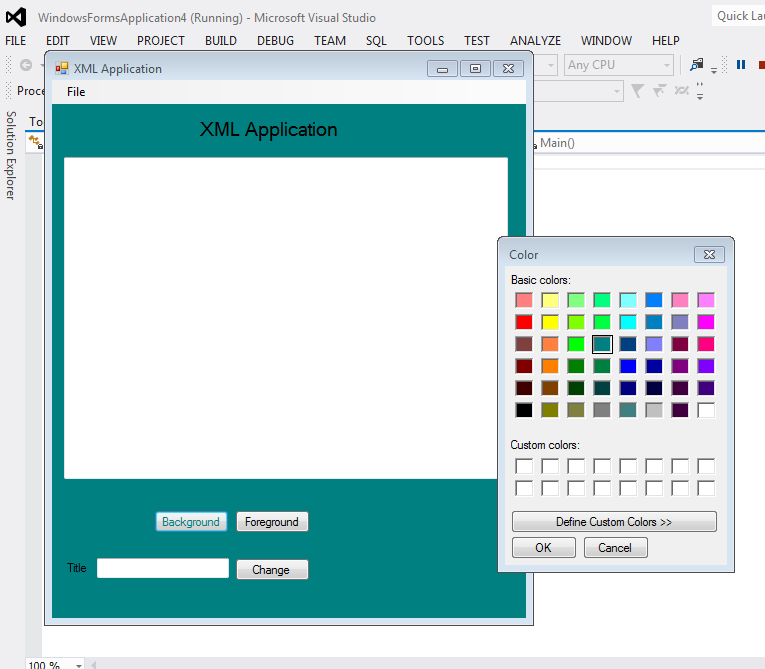
{

}

}

}





**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="300" Width="425"

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 Application1

{

class Sudent

{

public string id {set; get;}

public string name {set; get;}

public string age {set; get;}

}

}

**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:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections;

namespace ConsoleApplication6

{

class Program

{

static void Main(string[] args)

{

task2();

}

public static void task2()

{

List<student> l = new List<student>();

student s;

l.Add(s=new student());

s.name = "ALi";

s.subject = "Chemistry";

s.marks = 50;

l.Add(s=new student());

s.name = "ALi";

s.subject = "Biology";

s.marks = 60;

l.Add(s=new student());

s.name = "Chishti";

s.subject = "Chemistry";

s.marks = 50;

l.Add(s=new student());

s.name = "Chishti";

s.subject = "Biology";

s.marks = 70;

var results = from st in l

group st by st.name into g

select new {

name = g.Key,

Average = g.Average(c => c.marks)

};

foreach(var a in results)

{

Console.WriteLine(a);

}

Console.ReadLine();

}

public static void task1()

{

int[] r = new int[1000];

Random ran = new Random();

for (int i = 0; i < 1000; i++)

{

r[i] = ran.Next(100);

}

var result = r.Where(n => n % 2 == 1);

foreach (var n in result)

{

Console.WriteLine(n);

Console.ReadLine();

}

}

}

}

**Student Class:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication6

{

class student

{

public string name;

public string subject;

public int 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:**

