

Visual Programming Language

Lab Manual [Fall 2019]

Student Name: Khwaja Omer Hussain

Student ID: 15356

|  |  |
| --- | --- |
| 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("KhwajaOmer Hussain \n \_\_\_\_\_\_\_ \n Developer");

Console.WriteLine("\nEducation \National College

College" +

" Secondary School 2016 \nMatriculation: The Educational Centre" );

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

"CSS Languages");

Console.WriteLine("\nExperience: one years");

Console.WriteLine("\nContact \n000000000urt @hotmail.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");

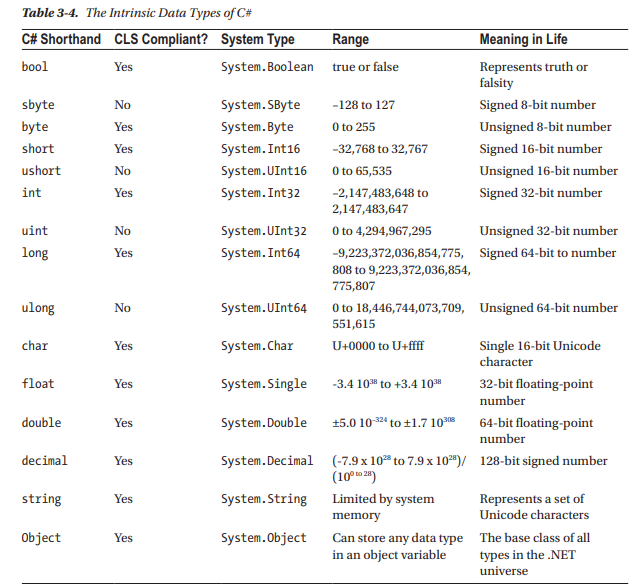
}

}

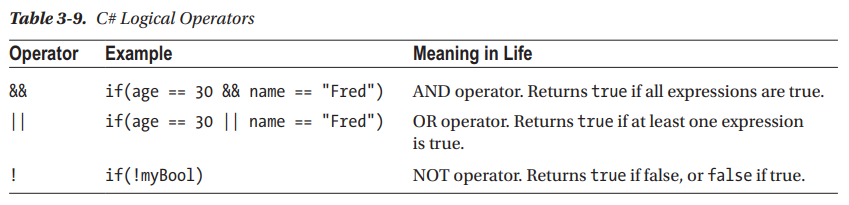
}

**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 lab2

{

using System;

public class Program

{

public static void Main()

{

int[] arr1 = new int[100];

int[] fr1 = new int[100];

int n, i, j, ctr;

Console.Write("\n\nFrequency of each element in an array:\n");

Console.Write("Input number of elements stored in an array: ");

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

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

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

{

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

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

fr1[i] = -1;

}

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

{

ctr = 1;

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

{

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

{

ctr++;

fr1[j] = 0;

}

}

if (fr1[i] != 0)

{

fr1[i] = ctr;

}

}

Console.Write("\nFrequency of all elements of an array: \n");

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

{

if (fr1[i] != 0)

{

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

}

}

}

}

}

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 lab2b

{

class Program

{

static void Main(string[] args)

{

int[] x = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

int max = x[0];

int min = x[0];

foreach (int v in x){

if(v > max) {

max = v;

}

if(v < min){

min = v;}

}

Console.WriteLine("maximum is {0}", max);

Console.WriteLine("mainimum is {0}", min);

}

}

}

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

**Code:**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab2c

{

class Program

{

static void Main(string[] args)

{

int[] x = { 10, 8, 10, 18, 17 };

ArrayList even = new ArrayList();

ArrayList odd = new ArrayList();

foreach (int v in x)

{

if (v % 2 == 0)

{

even.Add(v);

}

else

{

odd.Add(v);

}

}

int[] ea=new int[even.Count];

int[] oa=new int[odd.Count];

int i = 0;

Console.WriteLine("Even: ");

foreach (int v in even)

{

ea[i] = v;

Console.Write(" " + v);

i++;

}

i = 0;

Console.WriteLine("\nOdd: ");

foreach (int v in odd)

{

oa[i] = v;

Console.Write(" " + v);

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 lab2d

{

class Program

{

static void Main(string[] args)

{

string str;

int length = 0;

Console.Write("Input 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 lab2e

{

class Program

{

static void Main(string[] args)

{

string str;

int wrd,l;

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

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

Console.Write("Input string: ");

str = Console.ReadLine();

l = 0;

wrd = 1;

while (l <= str.Length - 1)

{

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

{

wrd++;

}

l++;

}

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

}

}

}

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 lab2f

{

using System;

public 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\n Calculate the Fibonacci number:\n");

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

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

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

}

}

}

**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.

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3

{

class Circle

{

private float r;

public float radius

{

get

{

Console.WriteLine("Get: ");

return r;

}

set

{

Console.WriteLine("Set: ");

this.r = value;

}

}

public void setRadius(float value)

{

this.r = value;

}

}

}

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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3b

{

class Circle

{

private float r;

public float radius

{

get {

Console.WriteLine("Get: ");

return r;

}

set {

if (this.r >0)

{

this.r = value;

}

else

{

throw new ArgumentException("Negative not allowed");

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3

{

class Program

{

static void Main(string[] args)

{

// Circle obj = new Circle();

//obj.radius = -1;

IMPAKTT House, Plot no# C-10,  opposite Mausoleum of Quaid-e-Azam Muhammad Ali Jinnah,

obj.name = "NTT";

obj.cgpa = 2.9f;

Console.WriteLine(obj.id);

Console.WriteLine(obj.name);

Console.WriteLine(obj.cgpa);

}

}

}

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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3c

{

class Circle

{

private Double radius;

public Double Radius

{

get {

return radius;

}

set {

if (value > 0){

radius = value;

}else{

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

}

}

}

public Circle(Double radius)

{

}

public Circle()

{

this.radius = 0.0;

}

public Double getArea(Double rad) {

return 3.14 \* (rad \* rad);

}

}

class Program

{

static void Main(string[] args)

{

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

Circle a = new Circle();

try {

a.Radius = val;

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

}

catch(ArgumentException ex){

Console.WriteLine(ex);

}

Console.ReadLine();

}

}

}

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

**Code:**

class Program

{

static void Main(string[] args)

{

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

Circle a = new Circle();

try {

a.Radius = val;

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

}

catch(ArgumentException ex){

Console.WriteLine(ex);

}

Console.ReadLine();

}

}

}

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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3d

{

class Circle

{

private Double radius;

public Double Radius

{

get {

return radius;

}

set {

if (value > 0){

radius = value;

}else{

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

}

}

}

public Circle(Double radius)

{

}

public Circle()

{

this.radius = 0.0;

}

public Double getArea(Double rad) {

return 3.14 \* (rad \* rad);

}

}

class Program

{

static void Main(string[] args)

{

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

Circle a = new Circle();

try {

a.Radius = val;

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

}

catch(ArgumentException ex){

Console.WriteLine(ex);

}

Console.ReadLine();

}

}

}

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

1. Inheritance/ Polymorphism
   * + 1. Create an abstract class Animal. Define a property: name of type string. Define an abstract method sound(). Define a constructor to initialize the animal’s name.

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3e

{

public abstract class Animal

{

String Name;

public String Name1

{

get { return Name; }

set { Name = value; }

}

public abstract void Sound();

}

public abstract class mamel : Animal{

}

public abstract class Non\_mamel : Animal

{

}

public class cat : mamel

{

public override void Sound()

{

Console.WriteLine("Meow");

}

}

public class goat : mamel

{

public override void Sound()

{

Console.WriteLine("mao");

}

}

public class fish : Non\_mamel

{

public override void Sound()

{

Console.WriteLine("baa");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class test

{

public static void Main(String[] args)

{

mamel obj = new cat();

obj.Sound();

Console.Read();

}

}

}

**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:**

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 lab4

{

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)

{

int id = int.Parse(txtid.Text);

string name = txtname.Text;

string email = txtemail.Text;

string password = txtpass.Text;

string country = cmbcountry.Text;

string date = dateTimePicker1.Text;

string gender;

if (rdomail.Checked)

{

gender = "Male";

}

else { gender = "Female"; }

MessageBox.Show(gender);

if (chk1.Checked)

{

MessageBox.Show("please fill values");

}

}

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

{

}

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

{

Form2 obj = new Form2();

obj.Show();

}

}

}

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

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

**Lab Tasks:**

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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab5

{

class student

{

public string id;

public string name;

public int age;

public float cgpa;

public string Student\_ID

{

get

{

Console.WriteLine("Get gender");

return gender;

}

set

{

Console.WriteLine("Set Gender");

gender = value;

}

}

public string Student\_Name

{

get

{

Console.WriteLine("Get Student Name");

return name;

}

set

{

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

name = value;

}

}

public int Student\_Age

{

get

{

Console.WriteLine("Get Student Age");

return age;

}

set

{

Console.WriteLine("Set Student Age" );

age = value;

}

}

public float Student\_Cgpa

{

get

{

Console.WriteLine("Get student GPA");

return cgpa;

}

set

{

Console.WriteLine("Set Student GPA");

cgpa = value;

}

}

}

}

**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:**

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("Students added successfully");

}

private void button5\_Click(object sender, EventArgs e)

{

if (current >= students.Count-1)

{

return;

}

else {

current++;

Student s = students[current];

textBox1.Text = s.id;

textBox2.Text = s.age;

textBox3.Text = s.marks;

if (s.gender == "male")

{

Male.Checked = true;

}

else

{

Female.Checked = true; } }

}

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

{

if (current <= 0)

{

return;

}

else {

current--;

Student s = students[current];

textBox1.Text = s.id;

textBox2.Text = s.age;

textBox3.Text = s.marks;

if (s.gender == "male")

{

Male.Checked = true;

}

else

{

Female.Checked = true;

}

}

}

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; } } }}

STUDENT CLASS:

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;

}}

**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 WindowsFormsApplication2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

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

{

XmlDocument d = new XmlDocument();

d.Load(openFileDialog1.FileName);

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

this.Text = l[0].InnerText;

l = d.GetElementsByTagName("Text");

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

l = d.GetElementsByTagName("Width");

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

l = d.GetElementsByTagName("Height");

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

l = d.GetElementsByTagName("Background");

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

l = d.GetElementsByTagName("Foreground");

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

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

}

}

}

}

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

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

**Lab Task:**

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

**Code:**

<Window x:Class="WpfApplication1.MainWindow"

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

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

Title="MainWindow" Height="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.

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication16

{

class Program

{

static void Main(string[] args)

{

int[] nums = GenerateLotsOfNumbers(100);

Console.WriteLine("Numbers");

foreach (var i in nums)

{

Console.Write(i + "");

}

Console.WriteLine("\n odd");

var result = from n in nums

where n % 2 != 0

select n;

int count = result.Count();

int max = result.Max();

int min = result.Min();

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

foreach(var i in result)

{

Console.Write(i +"");

}

Console.ReadKey();

}

private static int[] GenerateLotsOfNumbers(int count)

{

Random generator = new Random(0);

int[] result = new int[count];

for (int i = 0; i < count; i++)

{

result[i] = generator.Next();

}

return result;

}

}

}

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

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication17

{

class Program

{

static void Main(string[] args)

{

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

stds.Add(new Student ("A", "English",89));

stds.Add(new Student("B", "English", 78));

stds.Add(new Student("C", "English", 51));

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

foreach(var v in results)

{

Console.WriteLine(v);

}

Console.ReadKey();

}

}

}

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

**Code:**

student class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication17

{

class Student

{

public string name { set; get; }

public string subject { set; get; }

public int marks { set; get; }

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

{

this.name = name;

this.subject = subject;

this.marks = marks;

}

}

}

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

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

**Lab Tasks:**

1. Create a database named School. Create a table Student.
2. Now create a windows form and show all the students records in a Grid View
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

{

Interaction logic for MainWindow.xaml

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 = "365";

s.name = "omer";

s.age = 22;

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

s.id = "222";

s.name = "ali";

s.age = 23;

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

s.id = "737";

s.name = "osama";

s.age = 24;

datagrid1.ItemsSource = stds;

}

}

}