###### **Cairo University**

**Faculty of Computers and Information Information Systems Department Database Systems 1**

**Lab 8 (Introduction to C# and Windows Form Applications)**

In this the following goals will be achieved:

1. C# programming language is introduced
2. Creating C# console application using visual studio 2008
3. Creating windows forms application using visual studio 2008

# Introduction

C# (programming language) (pronounced as see sharp) is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, procedural, generic, object-oriented (class-based), and component-oriented programming disciplines.

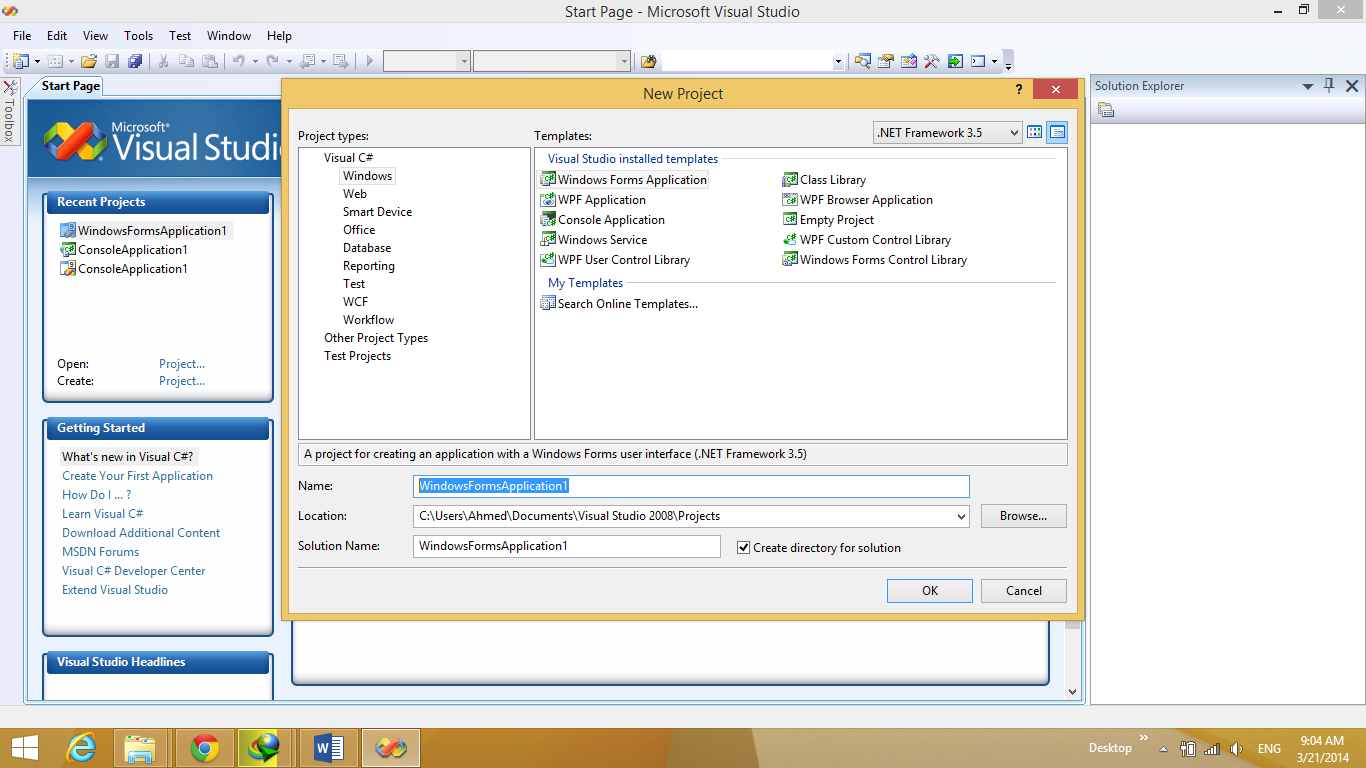
It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C♯ is one of the programming languages designed for the Common Language Infrastructure. C♯ is built on the syntax and semantics of C++, allowing C programmers to take advantage of .NET and the common language runtime.

C♯ is intended to be a simple, modern, general-purpose, object-oriented programming language.[6] Its development team is led by Anders Hejlsberg. The most recent version is C♯ 5.0, which was released on August 15, 2012.

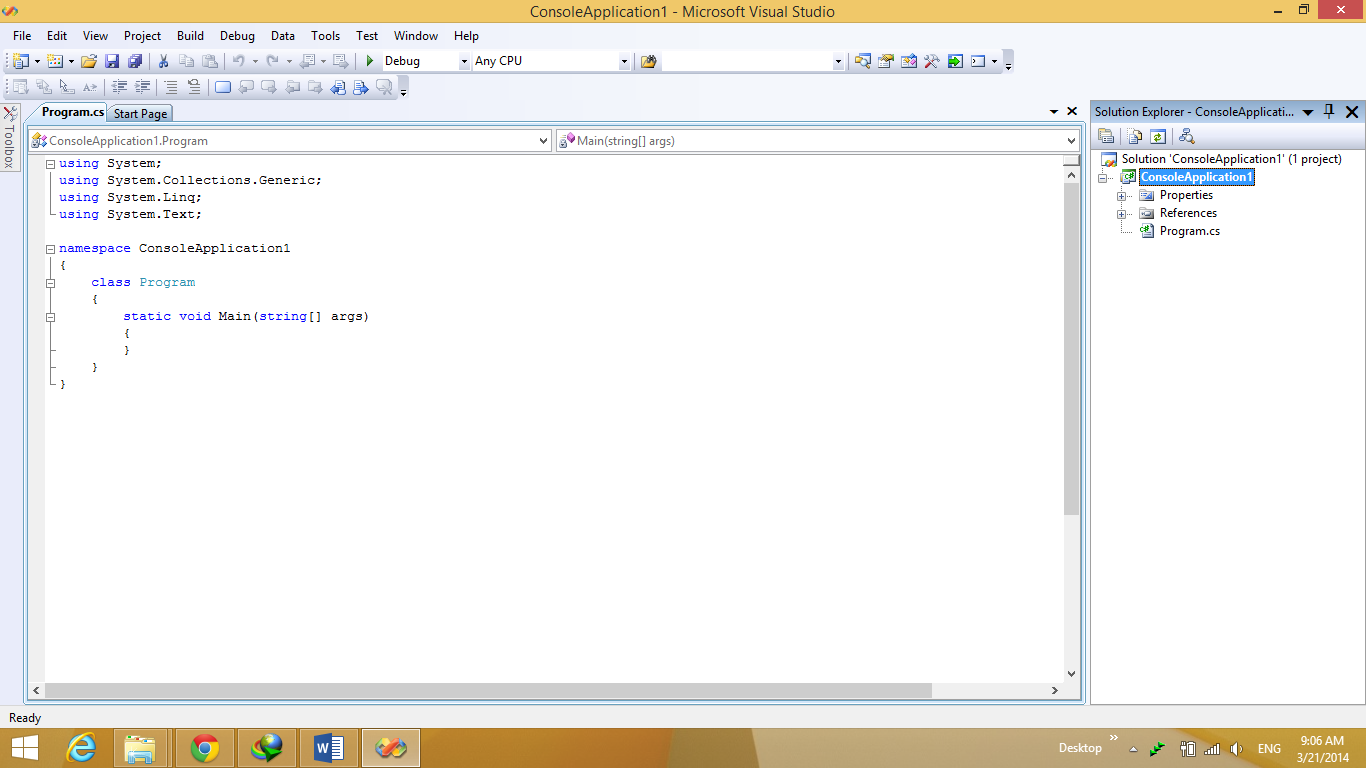
# Creating C# console application using visual studio 2008

To do these follow the following steps:

1. Open Microsoft visual studio 2008 application
2. File > New Project > Visual C# > Windows > Console application > and write down the project name.



1. You will find the following screen



1. The project tree is on the right side of your screen, your application starting point is “Program.cs” which you could consider holding the “main” function as in your c++ project.
2. Try writing the following code snippet “

Console.Out.WriteLine("Hello World this is test 1");”

And run them by pressing “ctrl+F5”.

1. Try more code loop for example “

int i = 0;

for (i = 0; i < 11; i++ )

{

if (i == 5) {

Console.Out.WriteLine("Reached the middle value");

} else {

Console.Out.WriteLine("I value is: " + i);

}

}

1. Next we will create a class called “Calculate” and instantiate a new object from it and run it by doing the following:
   1. Right click on the project name > add > class , now write down the class name.
   2. The class is now created write the following snippet code in the class body “

publicstaticdouble add(double num1, double num2) {

return num1 + num2;

}

”

1. Now return to “program.cs” and write the following code in order to test the class we have just created”

Console.Out.WriteLine("calc add test: " + Calculate.add(10.5, 36.7));

”

1. String is very powerful object and we will use it later a lot you can create strings and convert them to integers by as the following snippet”

string number = "100";

int num\_value = Int32.Parse(number);

num\_value++;

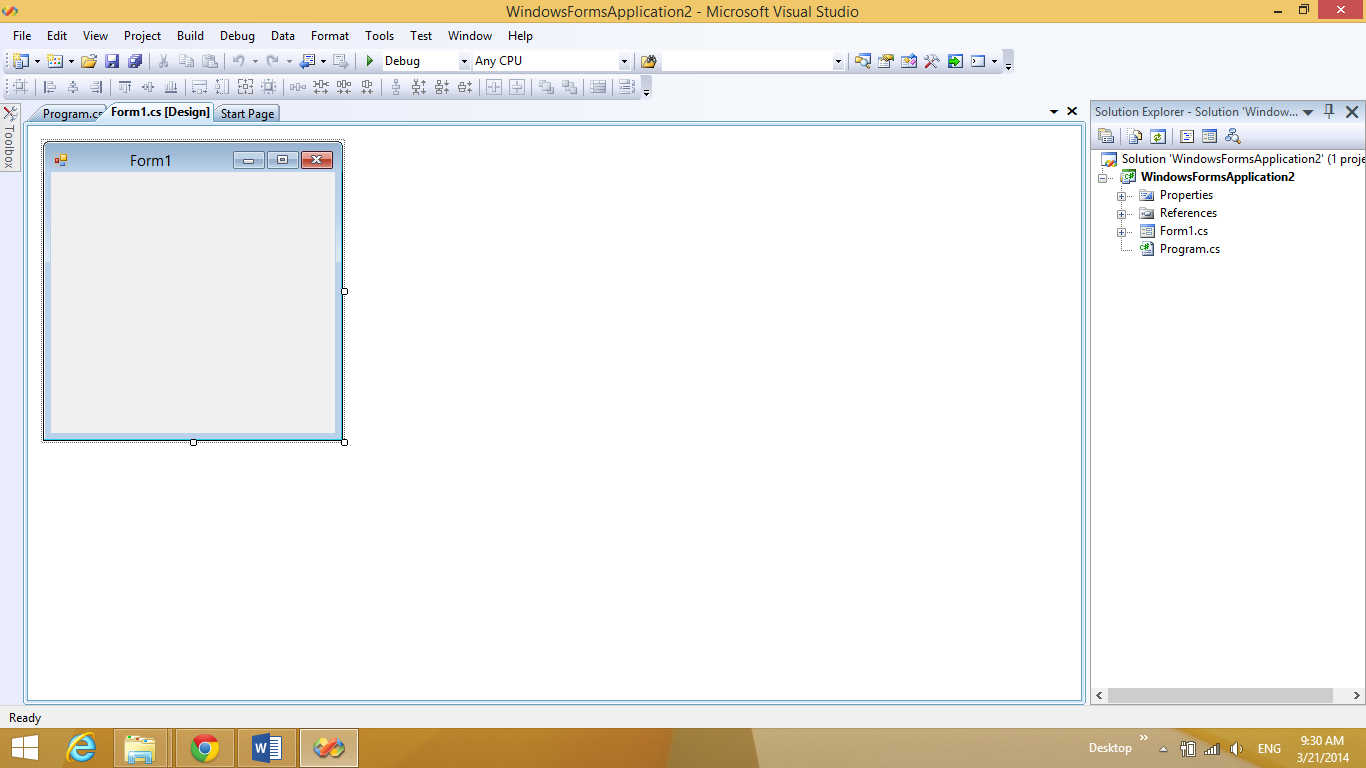
Console.Out.WriteLine("Number value is: " + num\_value);

”.

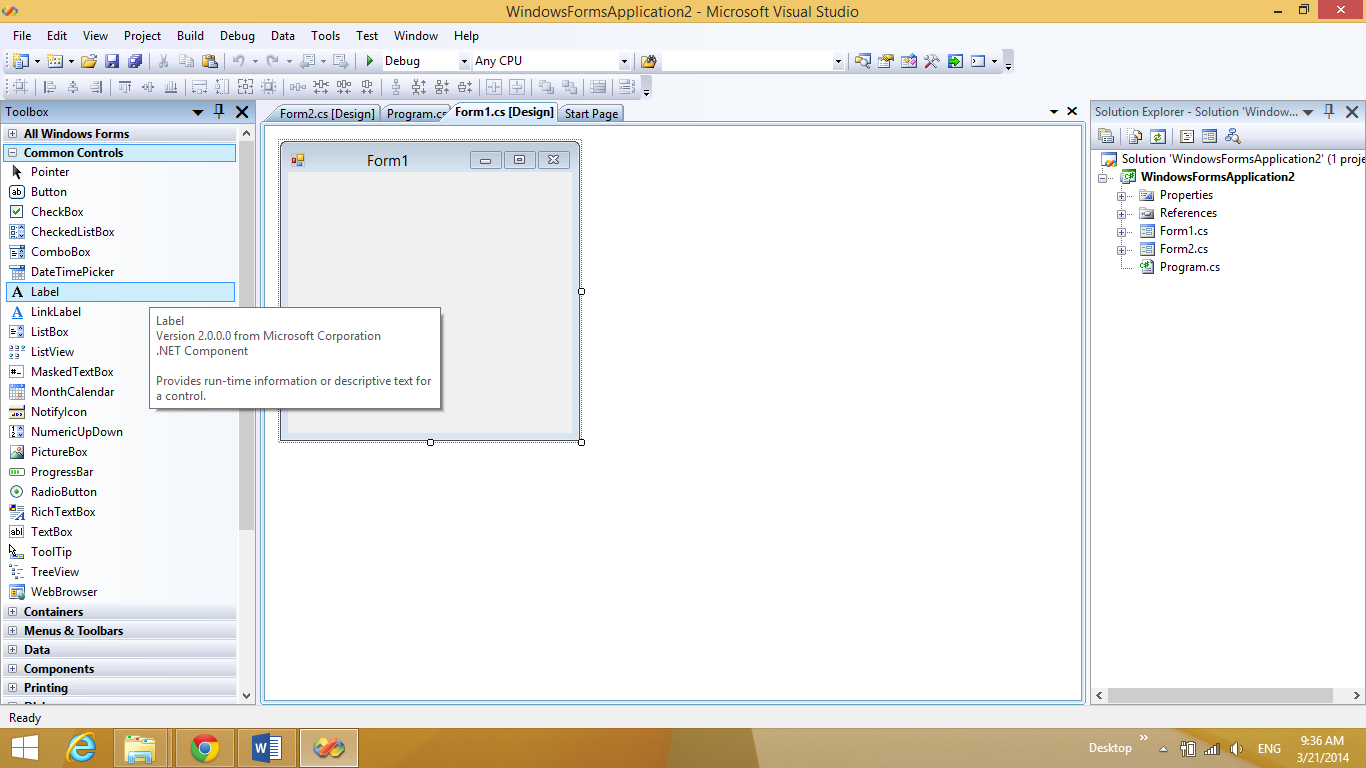
# Creating windows forms application using visual studio 2008

To do these follow the following steps:

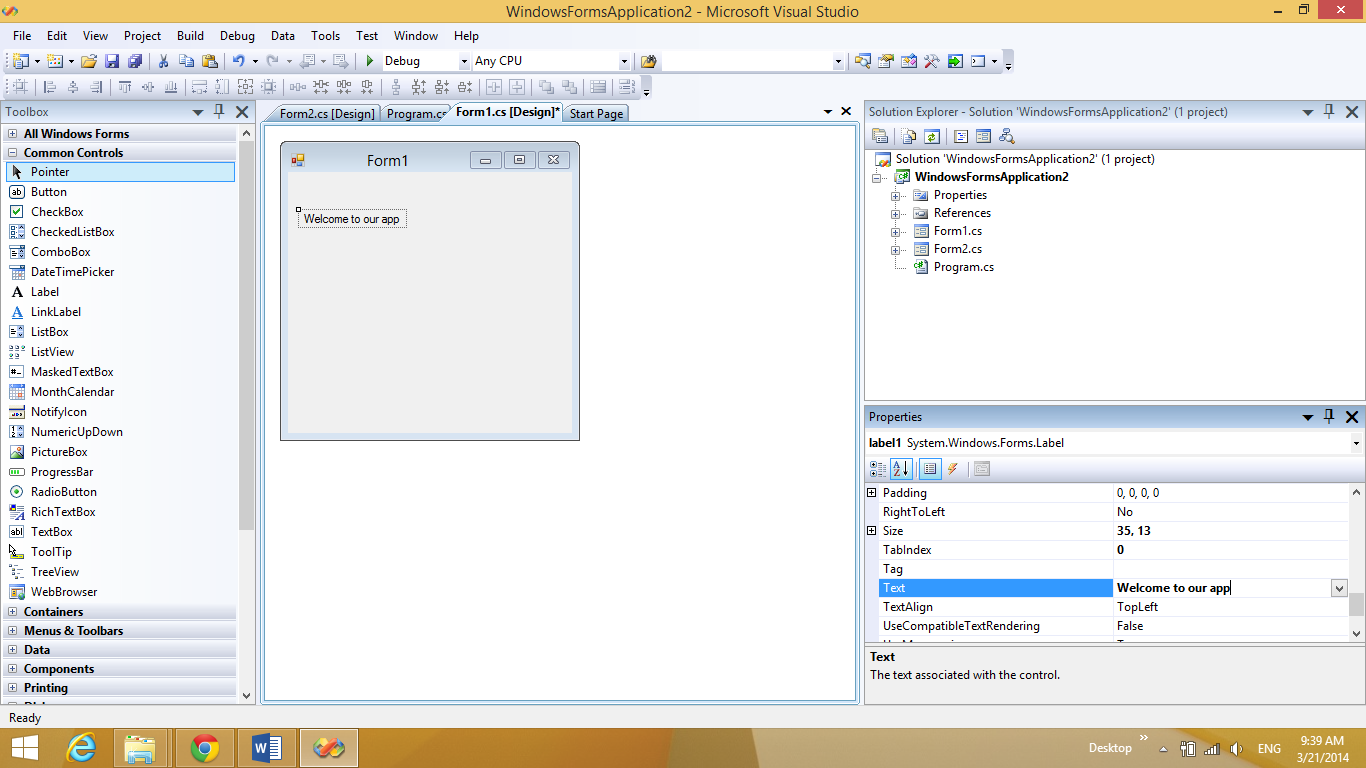
1. Open Microsoft visual studio 2008 application
2. File > New Project > Visual C# > Windows >WindowsFormsApplication> and write down the project name.
3. The project tree is on the right side of your screen, your application starting point is “Program.cs” which you could consider holding the “main” function as in your c++ project.
4. You will find a file created called “Form1” and it represent the Form UI appearing in your left side of screen.



1. The first form which will appear to you on running is “Form1”
2. To add more forms right click on your project > Add > Windows Form and write down the form name.
3. To change the first form to appear go to “program.cs” and change “Form1” in this line “Application.Run(newForm1());” to your desired form name.
4. First task to implement is to create a form containing welcome text do this by following these steps:
   1. Click on “toolbox” drag “Label” and release on your form

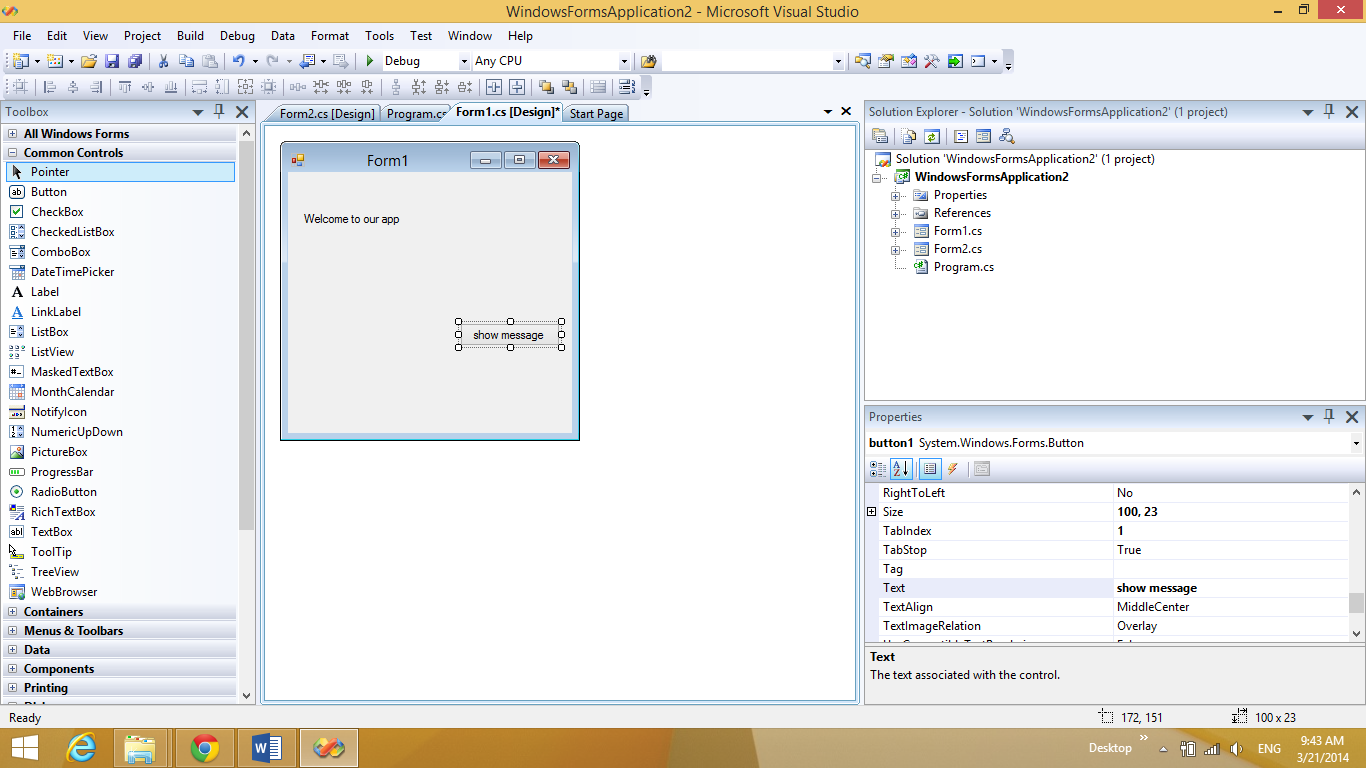


* 1. Click on the label and change its text from the properties window appearing

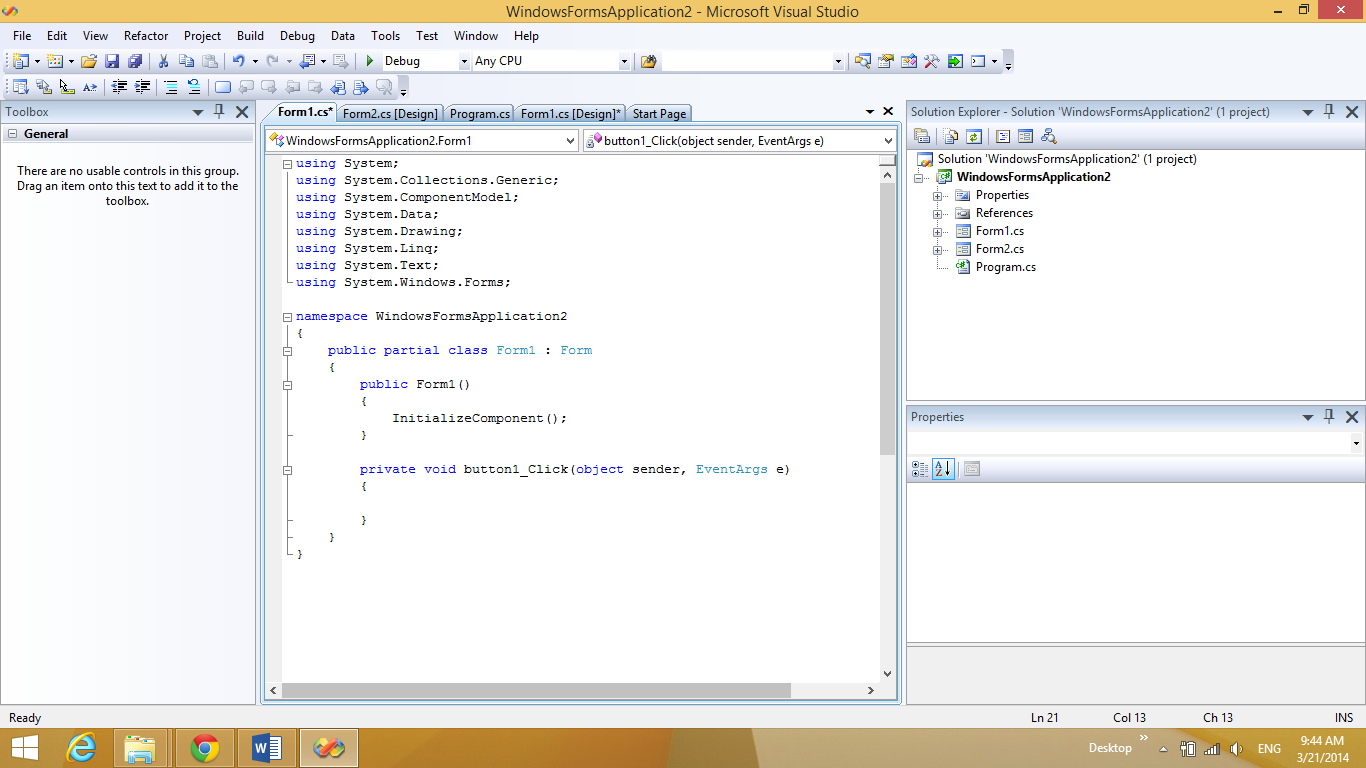


* 1. Try running now by pressing “ctrl+F5”.

1. Next task is to add a button and control its clicking action to display a message box by doing the following:
   1. Click on “toolbox” drag “Button” and release on your form
   2. Click on the button and change its text from the properties window appearing



* 1. Now double click the button and the following window will appear



* 1. Write the following code snippet in order to show the message box”

String message = "This is My firsr app";

MessageBox.Show(message);

“

* 1. Run now and test it

1. We now want to navigate from a form to another form do this by the following steps:
   1. Create a button like the previous task
   2. Write the following code snippet in order to show the next form”

// creating object of the new form

Form1 f1 = newForm1();

// hiding the current form

this.Hide();

//showing the created form

f1.Show();

“.

* 1. Run now and test it.

1. Next task will be passing a parameter from a form and displaying it on the other do this by the following steps:
   1. Go to the first form and drag a TextBox component on it as well as a new button
   2. Go to the second form and create a Label on it
   3. Go to the second form cs file and add the following function to its code ”

publicvoid set\_text(string text){

this.label1.Text = text;

}

”.

* 1. Go to the first form double click the button used for navigation and write the following snippet code ”

// // creating object of the second form

Form1 f1 = newForm1();

this.Hide();

// getting textfield text

stringmessage = this.TextField1.Text;

// setting the text in next form label

f1.set\_text(text);

f1.Show();

”.

1. Final task will be creating a button on your final form of navigation to safely close the application by the following steps:
   1. Creating a button on your final form as in step 11 and write down the following snippet as its action ”

Application.Exit();

”.

Now you have finished learning an introductory lesson about C# by creating two types of projects “console, and Windows Forms” apps.