C# Hello World: First Console Application Program

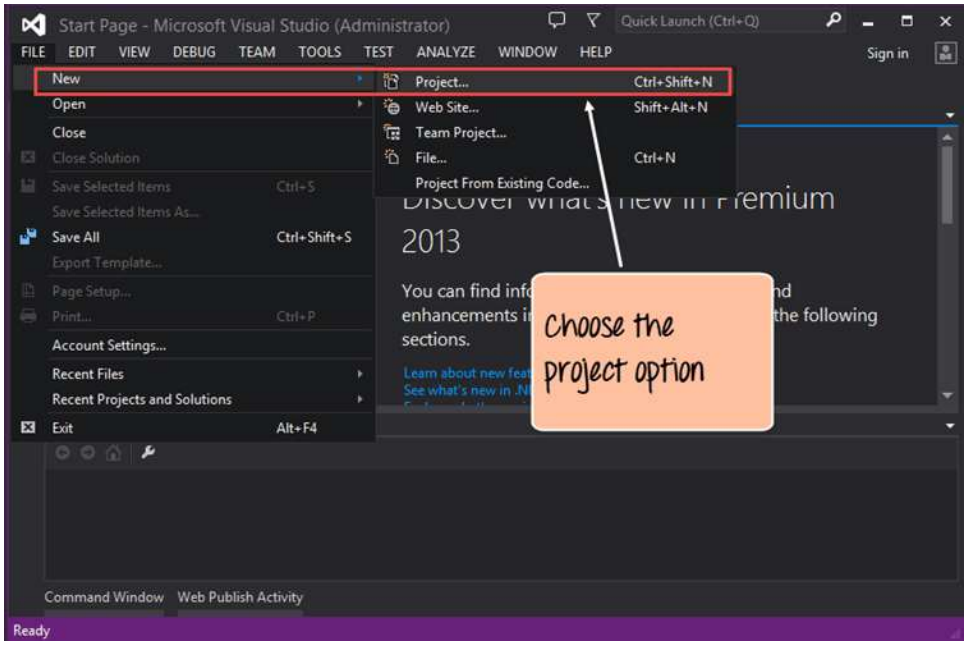
C# is one of the languages provided by Microsoft to work with .Net. This language encompasses a rich set of features, which allows developing different types of applications. C# is an object-oriented programming language and resembles several aspects of the C++ Language. In this lesson, we see how to develop our first application. This will be a basic console application.

Building the first console application

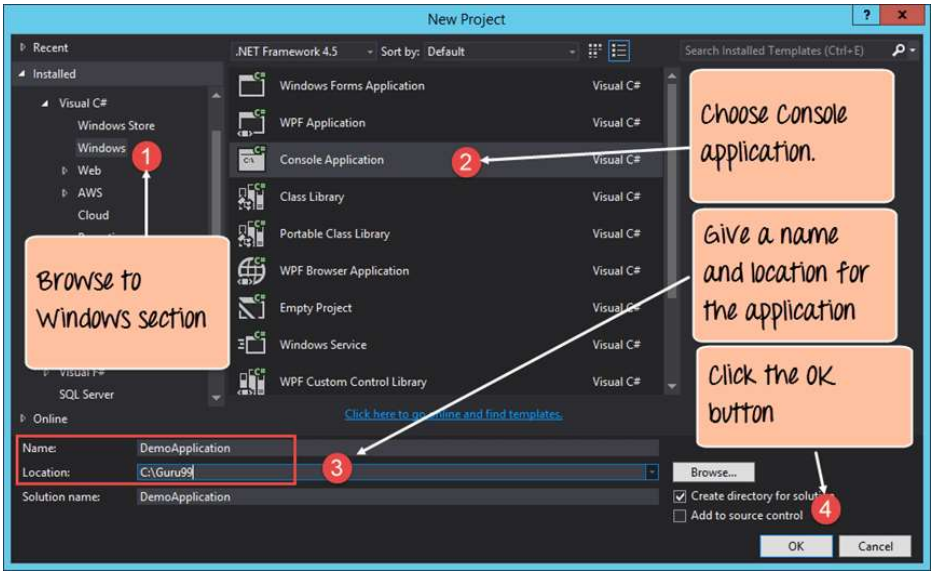
A console application is an application that can be run in the command prompt in Windows. For any beginner on .Net, building a console application is ideally the first step to begin with. In our example, we are going to use Visual Studio to create a console type project. Next, we are going to use the console application to display a message "Hello World". We will then see how to build and run the console application.

Let's follow the below mentioned steps to get this example in place.

Step 1) The first step involves the creation of a new project in Visual Studio. For that, once the Visual Studio is launched, you need to choose the menu option New->Project.



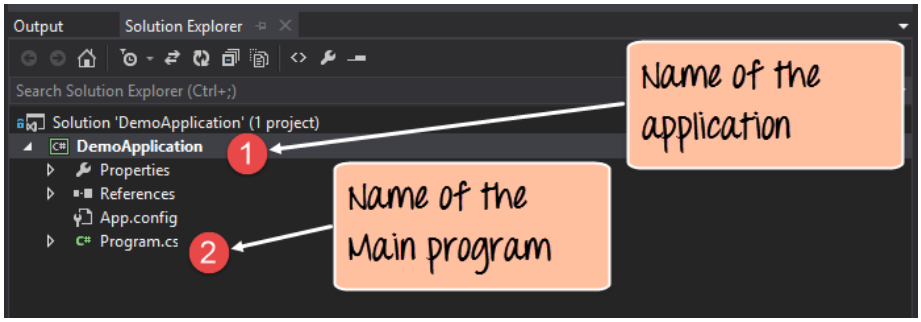
Step 2) The next step is to choose the project type as a Console application. Here, we also need to mention the name and location of our project.



1. In the project dialog box, we can see various options for creating different types of projects in Visual Studio. Click the Windows option on the left-hand side.
2. When we click the Windows options in the previous step, we will be able to see an option for Console Application. Click this option.
3. We then give a name for the application which in our case is DemoApplication. We also need to provide a location to store our application.
4. Finally, we click the 'OK' button to let Visual Studio to create our project.

If the above steps are followed, you will get the below output in Visual Studio.

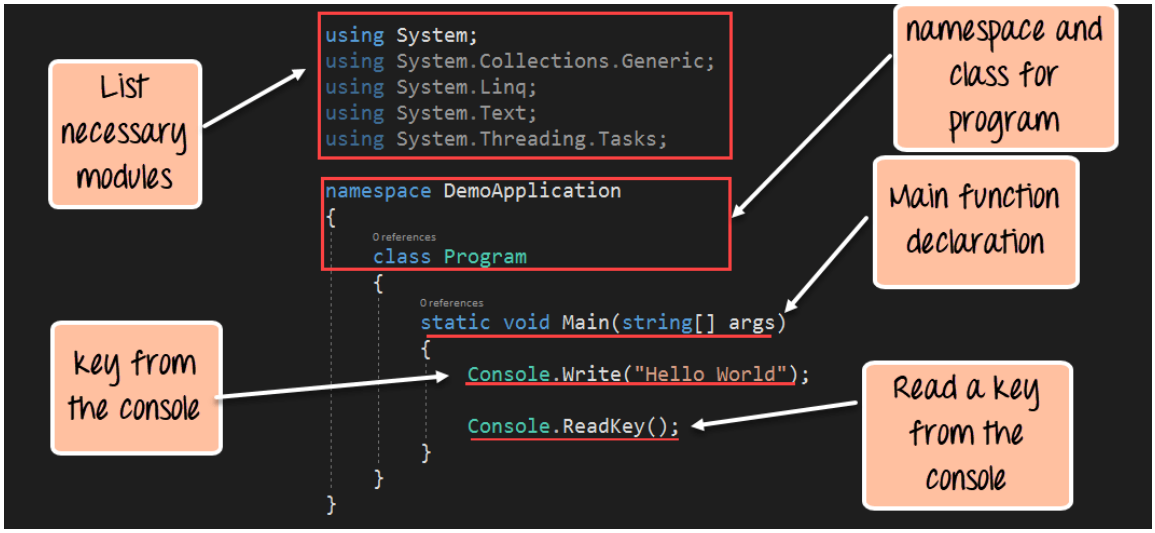
Output:-



1. A project called 'DemoApplication' will be created in Visual Studio. This project will contain all the necessary artifacts required to run the Console application.
2. The Main program called Program.cs is default code file which is created when a new application is created in Visual Studio. This code will contain the necessary code for our console application.

Step 3) Now let's write our code which will be used to display the string "Hello World" in the console application.

All the below code needs to be entered into the Program.cs file. The code will be used to write "Hello World" when the console application runs.



C# Hello World Program

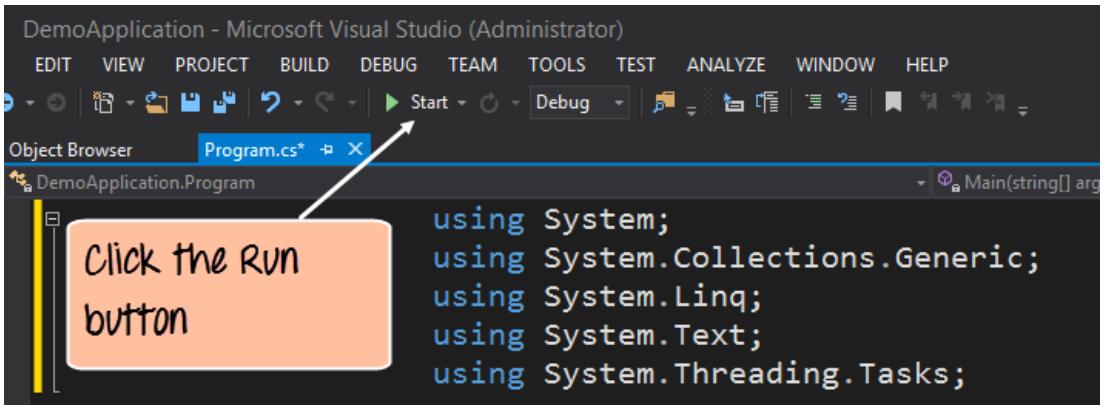
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace DemoApplication
{
 class Program
 {
  static void Main(string[] args)
  {
   Console.Write("Hello World");

   Console.ReadKey();
  }
 }
}
```

Code Explanation:-

1. The first lines of code are default lines entered by Visual Studio. The 'using' statement is used to import existing .Net modules in our console application. These modules are required for any .Net application to run properly. They contain the bare minimum code to make a code work on a Windows machine.

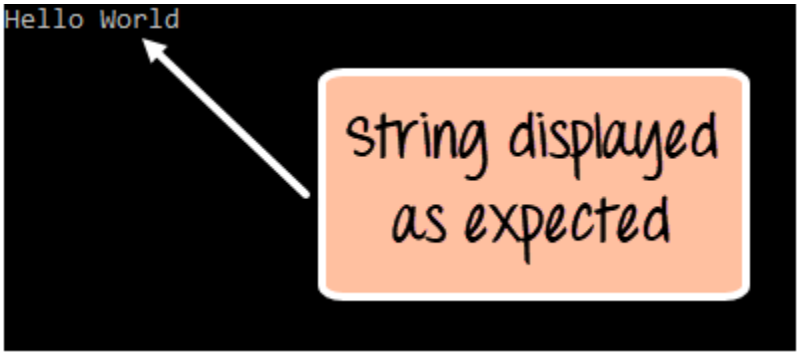
2. Every application belongs to a class. C# is an object-oriented language, and hence, all code needs to be defined in a self-sustaining module called a 'Class.' In turn, every class belongs to a namespace. A namespace is just a logical grouping of classes.
3. The Main function is a special function which is automatically called when a console application runs. Here you need to ensure to enter the code required to display the required string in the console application.
4. The Console class is available in .Net which allows one to work with console applications. Here we are using an inbuilt method called 'Write' to write the string "Hello World" in the console.
5. We then use the Console.ReadKey() method to read any key from the console. By entering this line of code, the program will wait and not exit immediately. The program will wait for the user to enter any key before finally exiting. If you don't include this statement in code, the program will exit as soon as it is run.

Step 4) Run your .Net program. To run any program, you need to click the Start button in Visual Studio.



If the above code is entered properly and the program is executed successfully, the following output will be displayed.

Output:



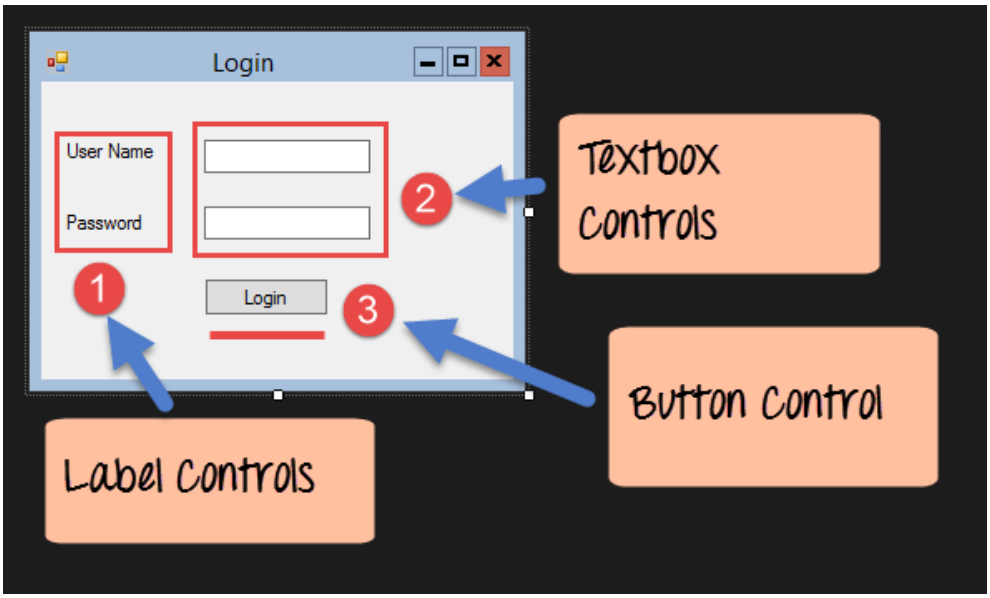
From the output, you can clearly see that the string "Hello World" is displayed properly. This is because of the Console.write statement causes this string to be sent to the console.

C# Windows Forms Application Tutorial with Example

In a real-life scenario team normally use Visual Studio and C# to create either Windows Forms or Web-based applications. A windows form application is an application, which is designed to run on a computer. It will not run on web browser because then it becomes a web application.

Windows Forms Basics

A Windows forms application is one that runs on the desktop computer. A Windows forms application will normally have a collection of controls such as labels, textboxes, list boxes, etc. Below is an example of a simple Windows form application C#. It shows a simple Login screen, which is accessible by the user. The user will enter the required credentials and then will click the Login button to proceed.



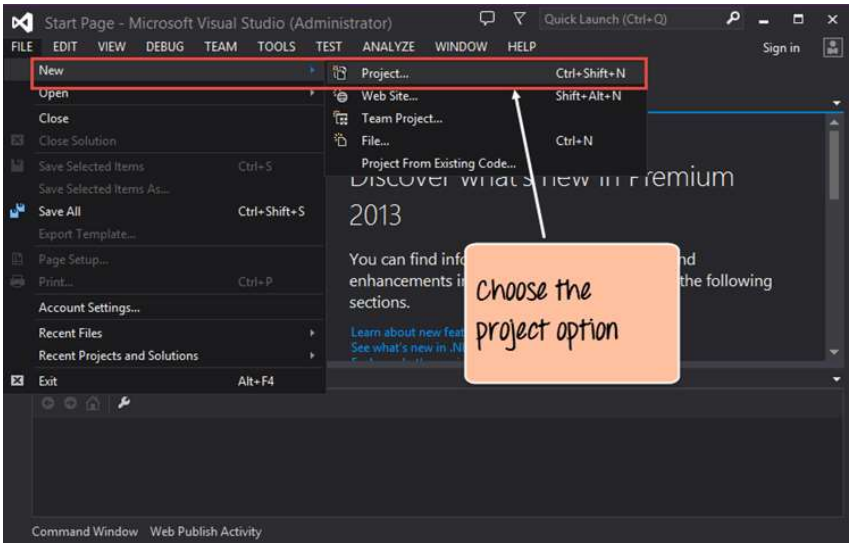
So an example of the controls available in the above application

1. This is a collection of label controls which are normally used to describe adjacent controls. So in our case, we have 2 textboxes, and the labels are used to tell the user that one textbox is for entering the user name and the other for the password.
2. The 2 textboxes are used to hold the username and password which will be entered by the user.
3. Finally, we have the button control. The button control will normally have some code attached to perform a certain set of actions. So for example in the above case, we could have the button perform an action of validating the user name and password which is entered by the user.

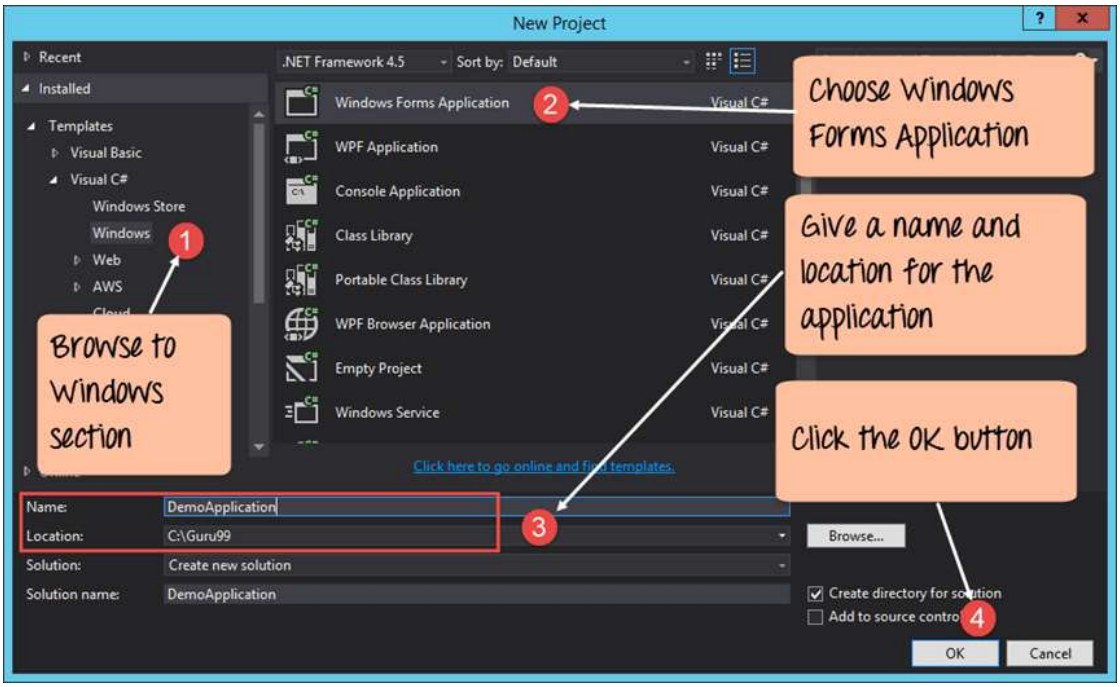
C# Hello World

Now let's look at an example of how we can implement a simple 'hello world' application in Visual Studio. For this, we would need to implement the below-mentioned steps

Step 1) The first step involves the creation of a new project in Visual Studio. After launching Visual Studio, you need to choose the menu option New->Project.



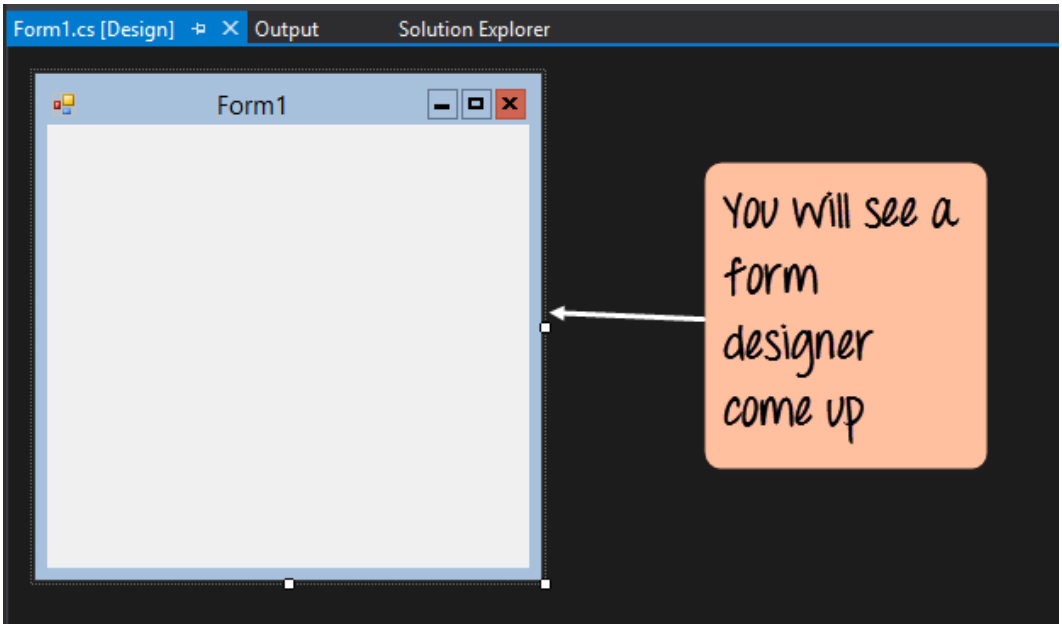
Step 2) The next step is to choose the project type as a Windows Forms application. Here we also need to mention the name and location of our project.



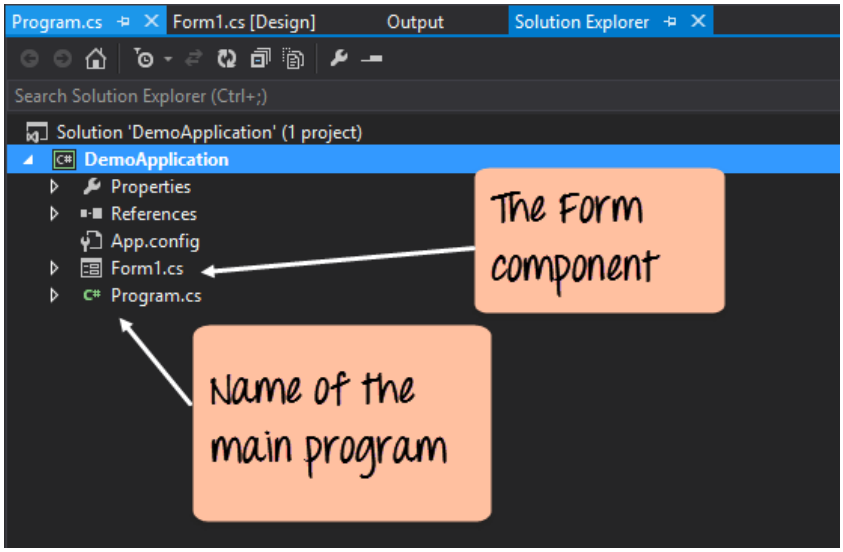
1. In the project dialog box, we can see various options for creating different types of projects in Visual Studio. Click the Windows option on the left-hand side.
2. When we click the Windows options in the previous step, we will be able to see an option for Windows Forms Application. Click this option.
3. We will give a name for the application. In our case, it is DemoApplication. We will also provide a location to store our application.
4. Finally, we click the 'OK' button to let Visual Studio create our project.

If the above steps are followed, you will get the below output in Visual Studio.

Output:-



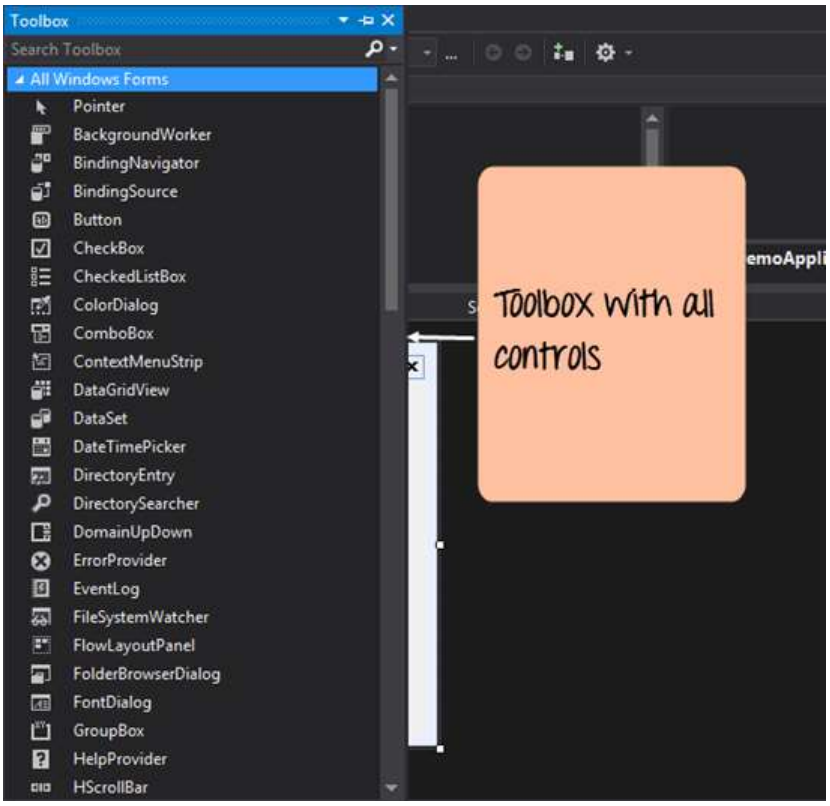
You will see a Form Designer displayed in Visual Studio. It's in this Form Designer that you will start building your Windows Forms application.



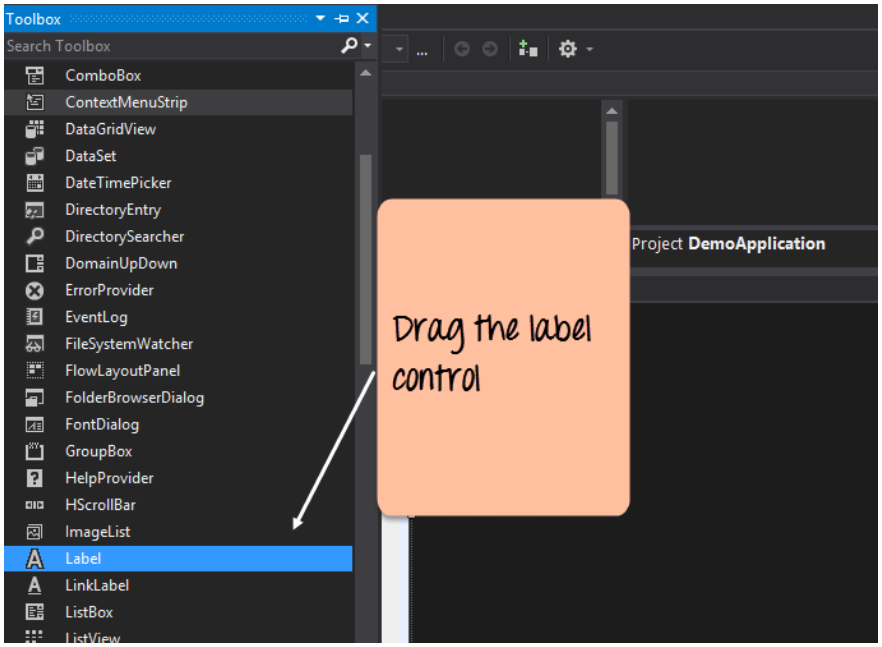
In the Solution Explorer, you will also be able to see the DemoApplication Solution. This solution will contain the below 2 project files

1. A Form application called Forms1.cs. This file will contain all of the code for the Windows Form application.
2. The Main program called Program.cs is default code file which is created when a new application is created in Visual Studio. This code will contain the startup code for the application as a whole.

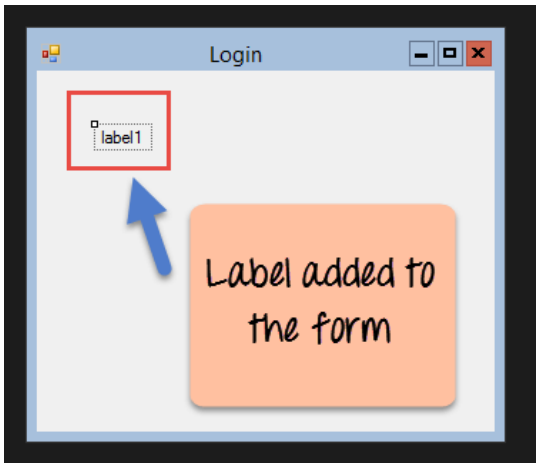
On the left-hand side of Visual Studio, you will also see a ToolBox. The toolbox contains all the controls which can be added to a Windows Forms. Controls like a text box or a label are just some of the controls which can be added to a Windows Forms. Below is a screenshot of how the Toolbox looks like.



Step 3) In this step, we will now add a label to the Form which will display "Hello World." From the toolbox, you will need to choose the Label control and simply drag it onto the Form.

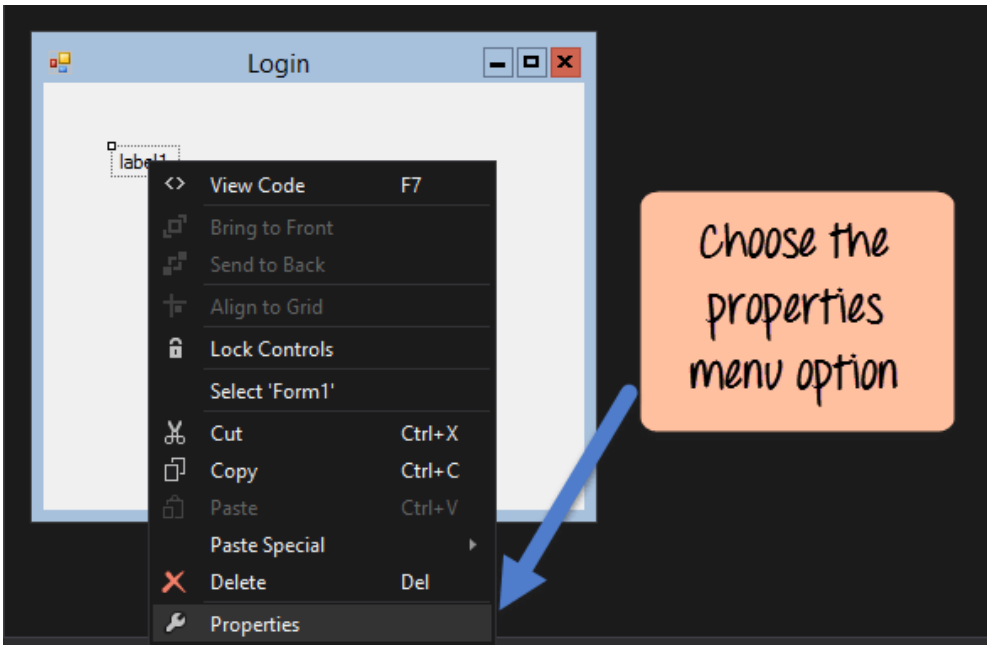


Once you drag the label to the form, you can see the label embedded on the form as shown below.

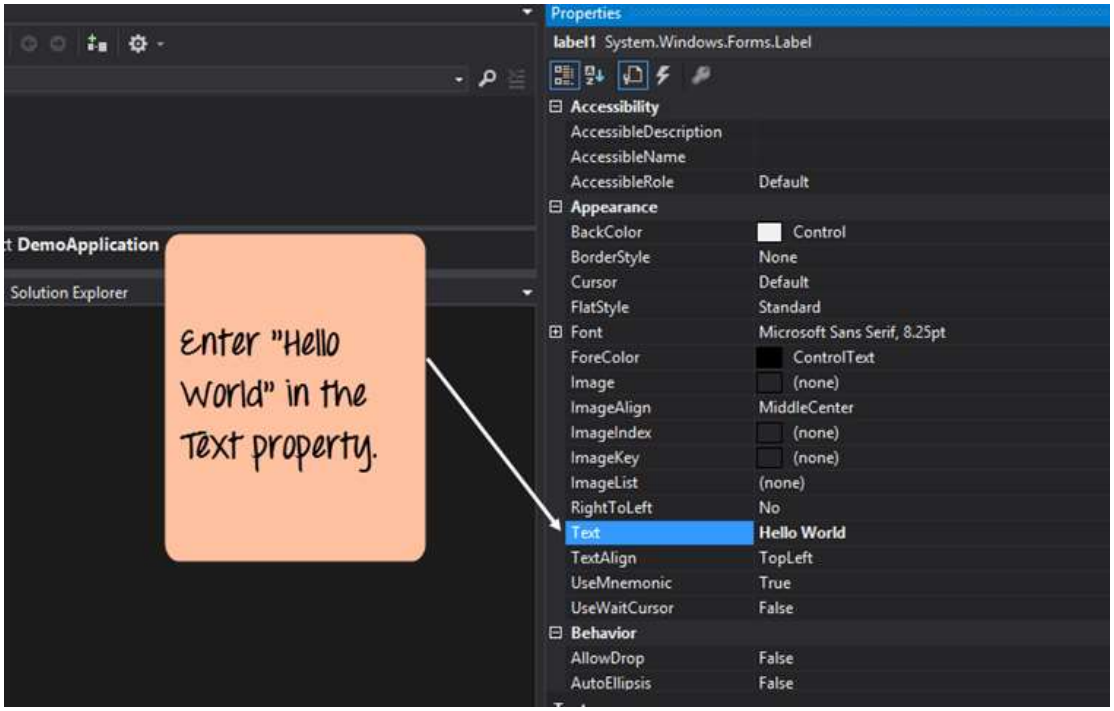


Step 4) The next step is to go to the properties of the control and Change the text to 'Hello World'.

To go to the properties of a control, you need to right-click the control and choose the Properties menu option

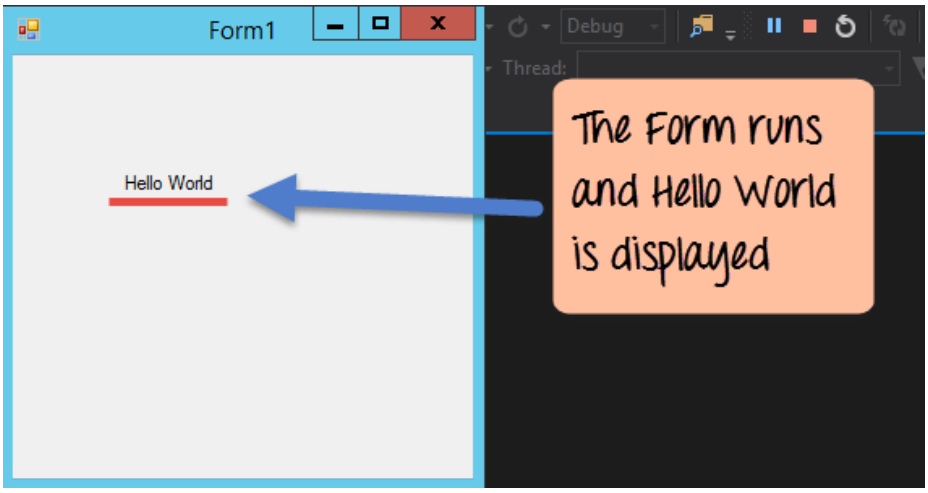


- The properties panel also shows up in Visual Studio. So for the label control, in the properties control, go to the Text section and enter "Hello World".
- Each Control has a set of properties which describe the control.



If you follow all of the above steps and run your program in Visual Studio, you will get the following output

Output:-



In the output, you can see that the Windows Form is displayed. You can also see 'Hello World' is displayed on the form.