



HNDIT1012 Visual Application Programming



Week 1



What is a program?

- A computer program is a sequence or set of instructions in a programming language for a computer to execute.
- Computer programs are one component of software.



Programming languages

- A computer programming language is a language used to write computer programs, which involves a computer performing some kind of computations.
- Thousands of different programming languages have been created, and more are being created every year.
- Many programming languages are written in an imperative form (i.e., as a sequence of operations to perform) while other languages use the declarative form (i.e. the desired result is specified, not how to achieve it).
- Eg: C++, C#, VisualBasic, Java etc..

Source code

- A computer program in its human-readable form is called source code.
- Source code needs another computer program to execute because computers can only execute their native machine instructions.
- Therefore, source code may be translated to machine instructions using the language's translators (compiler / Interpreter)

```
1  using System;
2
3  namespace CSharp_If_Statement
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              string name;
10             Console.WriteLine("Enter a name:");
11             name = Console.ReadLine();
12             if (name == "John");
13             {
14                 Console.WriteLine("Hi John!");
15             }
16             Console.ReadKey();
17         }
18     }
19 }
```

Executable file

- An EXE file is an executable program you can run in Microsoft Windows. Most EXE files contain either Windows applications or application installers.

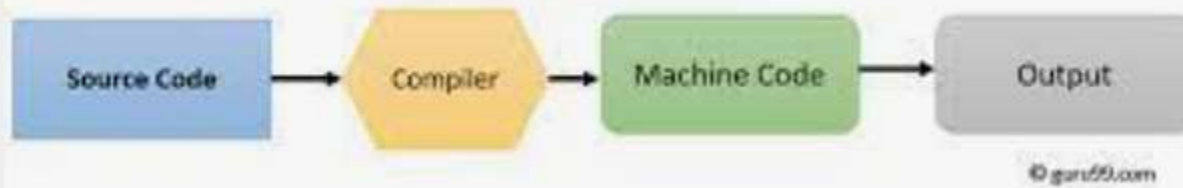




Language Translators

- Language translators allow computer programmers to write sets of instructions in specific programming languages. These instructions are converted by the language translator into machine code. The computer system then reads these machine code instructions and executes them.
 - Compiler
 - Interpreter

How Compiler Works



How Interpreter Works





NET Framework

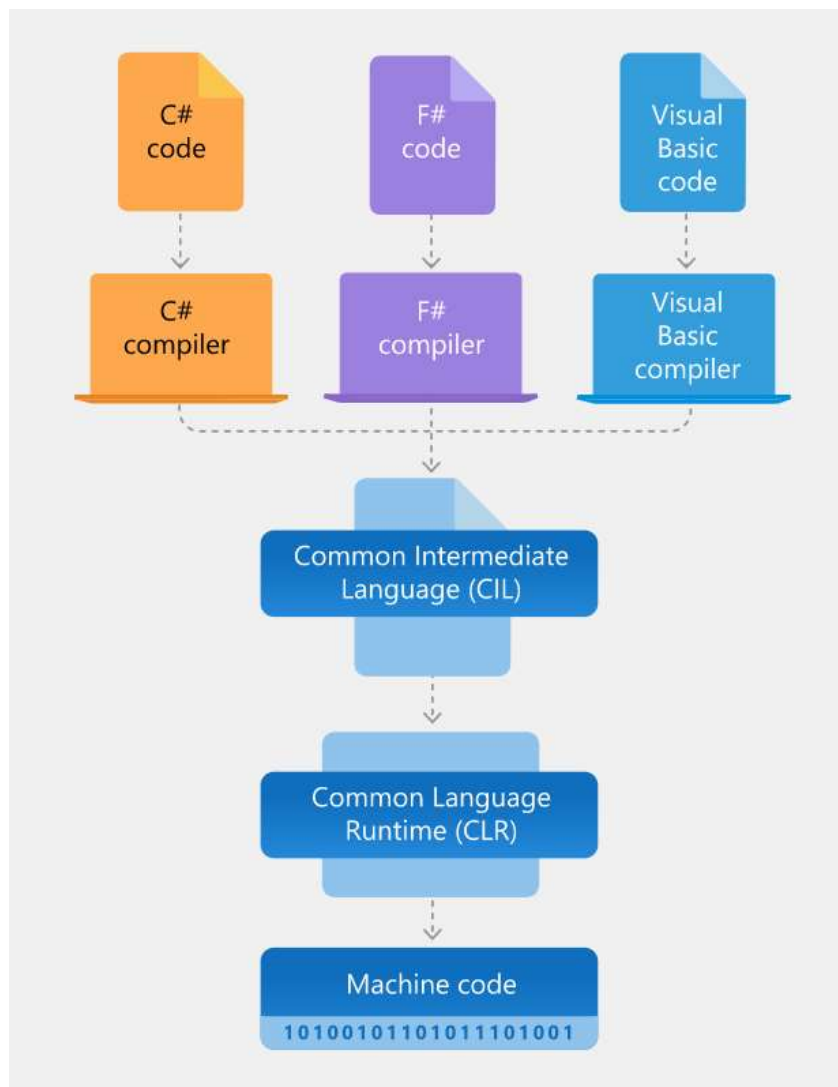
- NET Framework is a software development framework for building and running applications on Windows. . NET Framework is part of the . NET platform, a collection of technologies for building apps for Linux, macOS, Windows, iOS, Android, and more.



Architecture of .NET Framework ..

- .NET applications are written in the C#, F#, or Visual Basic programming language. Code is compiled into a language-agnostic Common Intermediate Language (CIL).
- Compiled code is stored in assemblies—files with a .dll or .exe file extension.
- When an app runs, the CLR takes the assembly and uses a just-in-time compiler (JIT) to turn it into machine code that can execute on the specific architecture of the computer it is running on.

Architecture of .NET Framework





Introduction to IDE (Visual studio 2022)

- An integrated development environment (IDE) is software for building applications that combines common developer tools into a single GUI.
- Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.



Visual studio 2022..

- The Visual Studio integrated development environment is a creative launching pad that you can use to edit, debug, and build code, and then publish an app.
- An integrated development environment (IDE) is a feature-rich program that can be used for many aspects of software development.
- Visual Studio includes compilers, code completion tools, graphical designers, and many more features to ease the software development process.



Create a new project **Save your project** **Run your code** **Edit your code** **Manage files, projects, & solutions** **Send feedback** **Sign in**

Manage server resources **Add controls to your UI** **Manage your Azure resources**

WebApplication2 - Microsoft Visual Studio (Administrator)

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Debug Any CPU Microsoft Edge

Cloud Explorer

Microsoft Azure

Resource Types

Search for resources

- (Local)
- Data Lake Analytics
- (Local)
- Databases
- Storage Accounts
 - (Development)
 - Blob Containers
 - Queues
 - Tables
- Free Trial (@outl)
- Storage Accounts
 - athenastorage2354
 - Blob Containers

Actions Properties

Name	at
Type	M
Subscription	Fr
Resource Group	At
Location	wi
Primary Key	E/
Secondary Key	Gr

What do you like about this tool?

What don't you like or feel is missing?

AccountController.cs HomeController.cs ManageController.cs

WebApplication2

```
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127
```

```
//  
// POST: /Account/VerifyCode  
[HttpPost]  
[AllowAnonymous]  
[ValidateAntiForgeryToken]  
References  
public async Task<ActionResult> VerifyCode(VerifyCodeViewM  
{  
    if (!ModelState.IsValid)  
    {  
        return View(model);  
    }  
  
    // The following code protects for brute force attacks.  
    // If a user enters incorrect codes for a specified am  
    // will be locked out for a specified amount of time.  
    // You can configure the account lockout settings in I  
    var result = await SignInManager.TwoFactorSignInAsync(  
        rememberBrowser: model.RememberBrowser);  
    switch (result)  
    {  
        case SignInStatus.Success:  
            return RedirectToLocal(model.ReturnUrl);  
        case SignInStatus.LockedOut:  
            return View("LockedOut");  
    }  
}
```

Output

Show output from: Package Manager

IsDirty	: False
FileCount	: 1
Name	: jquery-1.10.2.intellisense.js
Collection	: System.__ComObject
Properties	: System.__ComObject
NTF	: System.__ComObject

Output Azure App Service Activity

Ready Ln 1 Col 1 Ch 1 INS

View output from running, debugging, deploying, and more

Collaborate on code projects with your team

Solution Explorer

Search Solution Explorer (Ctrl+)

Solution 'WebApplication2' (1 pr

- WebApplication2
 - Properties
 - References
 - App_Data
 - App_Start
 - Content
 - Controllers
 - AccountController.cs
 - HomeController.cs
 - ManageController.cs
 - fonts
 - Models

Team Explorer - Home

Search Work Items (Ctrl+)

Home | docs-archive-project

- Azure DevOps
 - docs-archive-...
<https://docs-archive-...>
- Project
 - Web Portal | Task Board
 - Changes
 - Branches

Publish



The image shows the Visual Studio IDE interface with several callout boxes highlighting key features:

- Create a new project:** Points to the **File** menu.
- Run your code:** Points to the **Start** button in the toolbar.
- Launch Live Share:** Points to the **Live Share** icon in the top right.
- Send feedback:** Points to the **Feedback** icon in the top right.
- Add controls to your UI:** Points to the **Cloud Explorer** sidebar.
- Manage your Azure resources:** Points to the **Cloud Explorer** sidebar.
- Manage files, projects, and solutions:** Points to the **Solution Explorer** sidebar.
- Collaborate on code projects with your team:** Points to the **Team Explorer** sidebar.

The central editor displays the following C# code:

```
1 using System;
2 using System.Runtime.CompilerServices;
3
4 [assembly: InternalsVisibleTo("QuickTest")]
5 namespace QuickDate
6 {
7     [InternalsVisibleTo("QuickTest")]
8     internal class Calendar
9     {
10         static void Main(string[] args)
11         {
12             DateTime now = GetCurrentDate();
13             Console.WriteLine($"Today's date is {now}");
14             Console.ReadLine();
15         }
16
17         [InternalsVisibleTo("QuickTest")]
18         internal static DateTime GetCurrentDate()
19         {
20             return DateTime.Now.Date;
21         }
22     }
23 }
```



Visual studio 2022..

- **Solution Explorer** (top right) lets you view, navigate, and manage your code files. Solution Explorer can help organize your code by grouping the files into solutions and projects.
- The **editor window** (center), where you'll likely spend a majority of your time, displays file contents. This is where you can edit code or design a user interface such as a window with buttons and text boxes.
- The **Output window** (bottom center) is where Visual Studio sends notifications such as debugging and error messages, compiler warnings, publishing status messages, and more. Each message source has its own tab.



Installing VisualStudio 2022

Before you begin installing Visual Studio:

- Check the system requirements. These requirements help you know whether your computer supports Visual Studio 2022.
- Apply the latest Windows updates. These updates ensure that your computer has both the latest security updates and the required system components for Visual Studio.
- Reboot. The reboot ensures that any pending installs or updates don't hinder your Visual Studio install.
- Free up space. Remove unneeded files and applications from your system drive by, for example, running the Disk Cleanup app.



System Requirements for Visual Studio 2022 - Hardware

- 1.8 GHz or faster 64-bit processor; Quad-core or better recommended. ARM processors are not supported.
- Minimum of 4 GB of RAM. Many factors impact resources used; we recommend 16 GB RAM for typical professional solutions.
- Windows 365: Minimum 2 vCPU and 8 GB RAM. 4 vCPU and 16 GB of RAM recommended.
- Hard disk space: Minimum of 850 MB up to 210 GB of available space, depending on features installed; typical installations require 20-50 GB of free space.
- We recommend installing Windows and Visual Studio on a solid-state drive (SSD) to increase performance.
- Video card that supports a minimum display resolution of WXGA (1366 by 768); Visual Studio will work best at a resolution of 1920 by 1080 or higher.

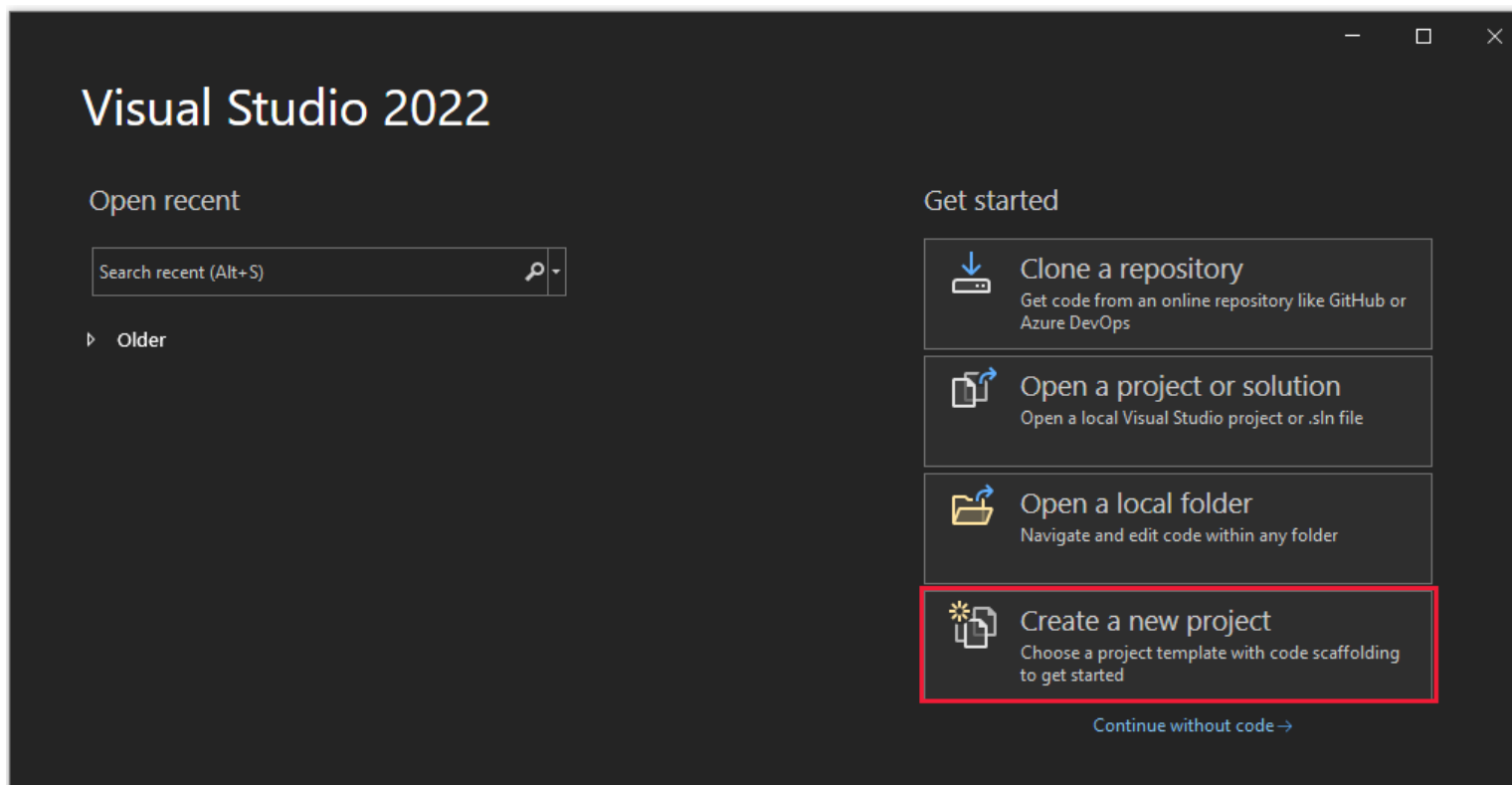


Download Visual Studio

- Download link
<https://visualstudio.microsoft.com/downloads>
- Follow the instructions in the following link.
<https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio?view=vs-2022>

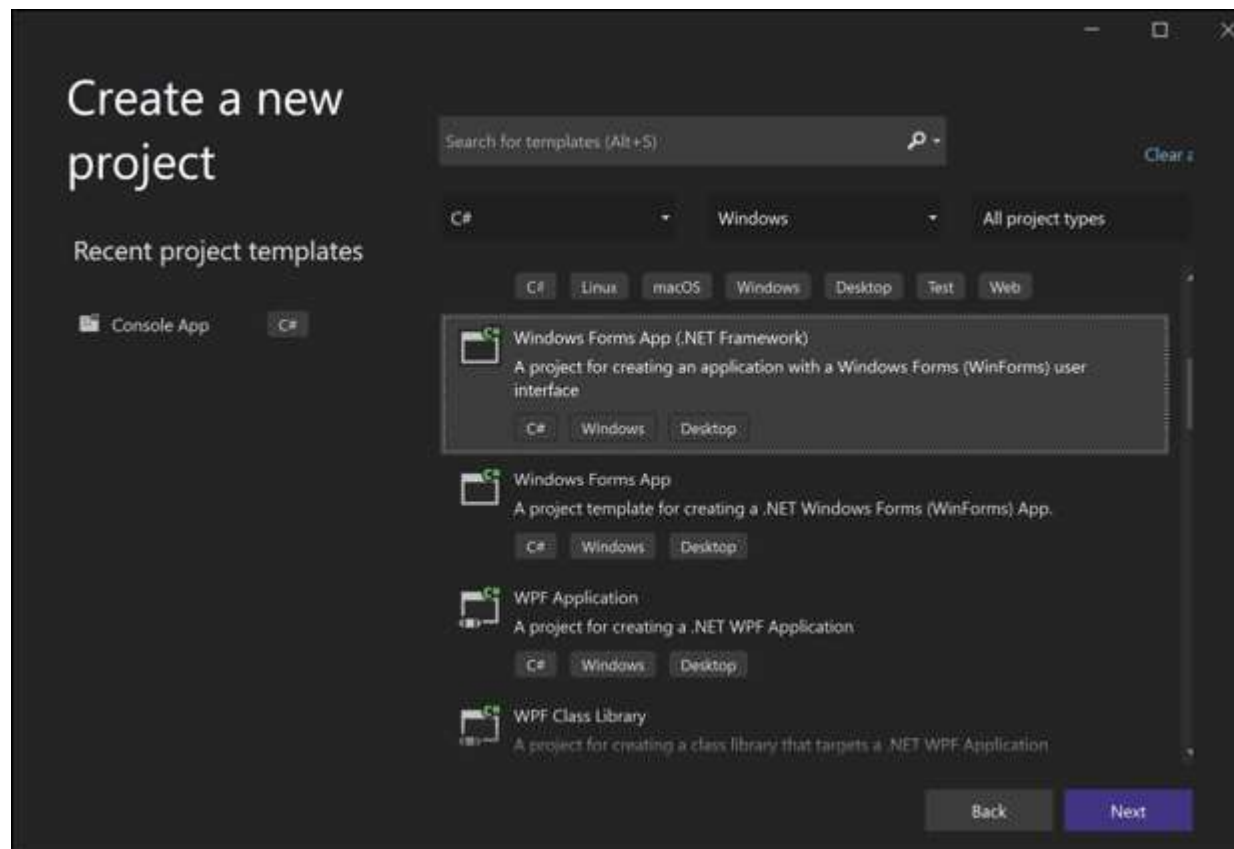
Creating a window based project.

- Open Visual Studio.
- On the start window, choose Create a new project.



Creating a window based project...

- On the Create a new project window, choose the Windows Forms App (.NET Framework) template for C#. ...





Creating a window based project...

- In the Configure your new project window, type or enter HelloWorld in the Project name box. Then, choose Create.

Configure your new project

Windows Forms App (.NET Framework) C# Windows Desktop

Project name
HelloWorld

Location
C:\Users\UserName\source

Solution name ⓘ
HelloWorld

☐ Place solution and project in the same directory

Framework
.NET Framework 4.7.2

Back Create

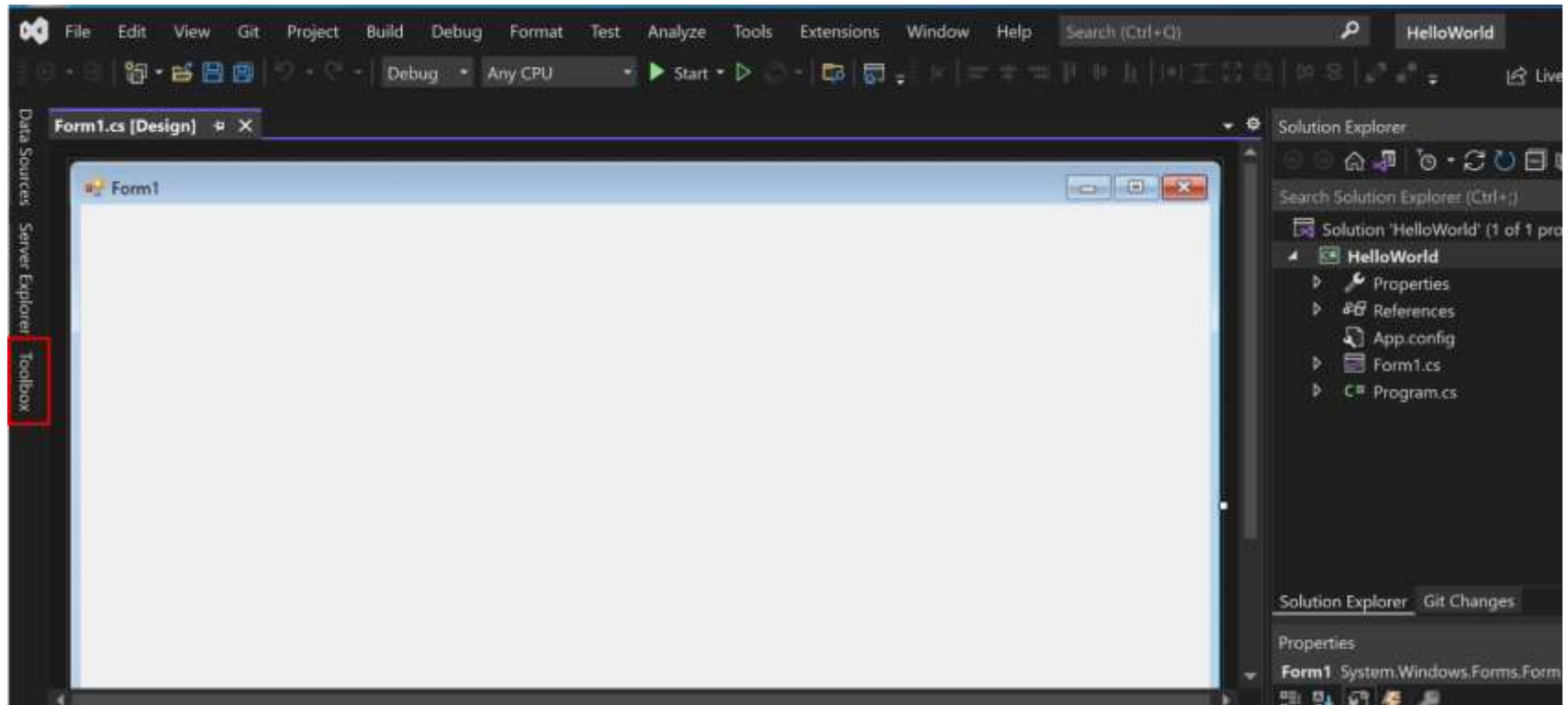


Create the application

- After you select your C# project template and name your file, Visual Studio opens a form for you. A form is a Windows user interface. We'll create a "Hello World" application by adding controls to the form, and then we'll run the app.

Create the application..

- Select Toolbox to open the Toolbox fly-out window.



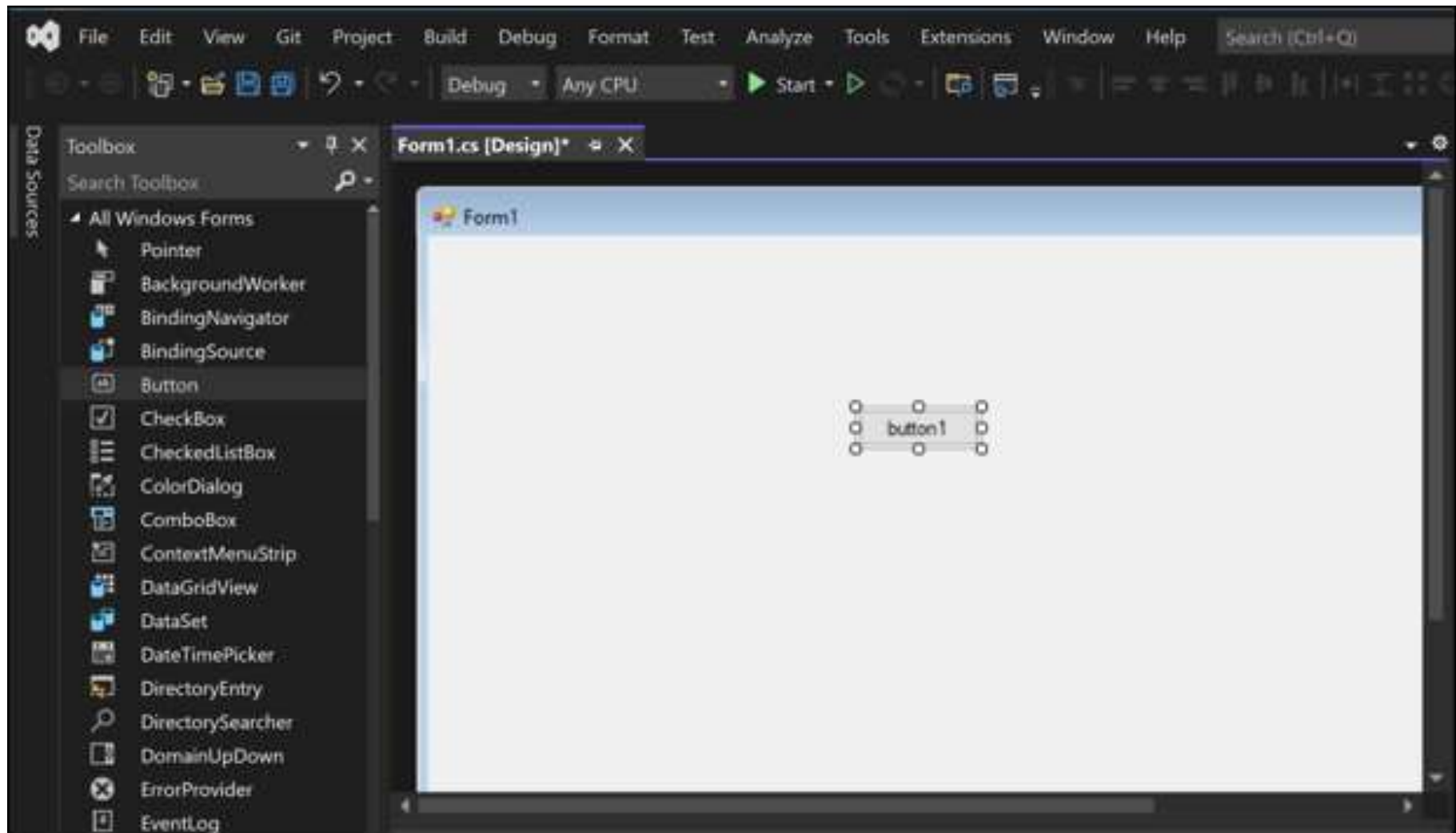


Create the application..

- Select the Pin icon to dock the Toolbox window.
- Select the Button control and then drag it onto the form.
- In the Properties window, locate Text, change the name from button1 to Click this, and then press Enter.
- In the Design section of the Properties window, change the name from button1 to btnClickThis, and then press Enter.



Create the application..





Create the application..

- Select the Label control from the Toolbox window, and then drag it onto the form and drop it beneath the Click this button.
- In either the Design section or the (DataBindings) section of the Properties window, change the name of label1 to lblHelloWorld, and then press Enter.



Add code to the form

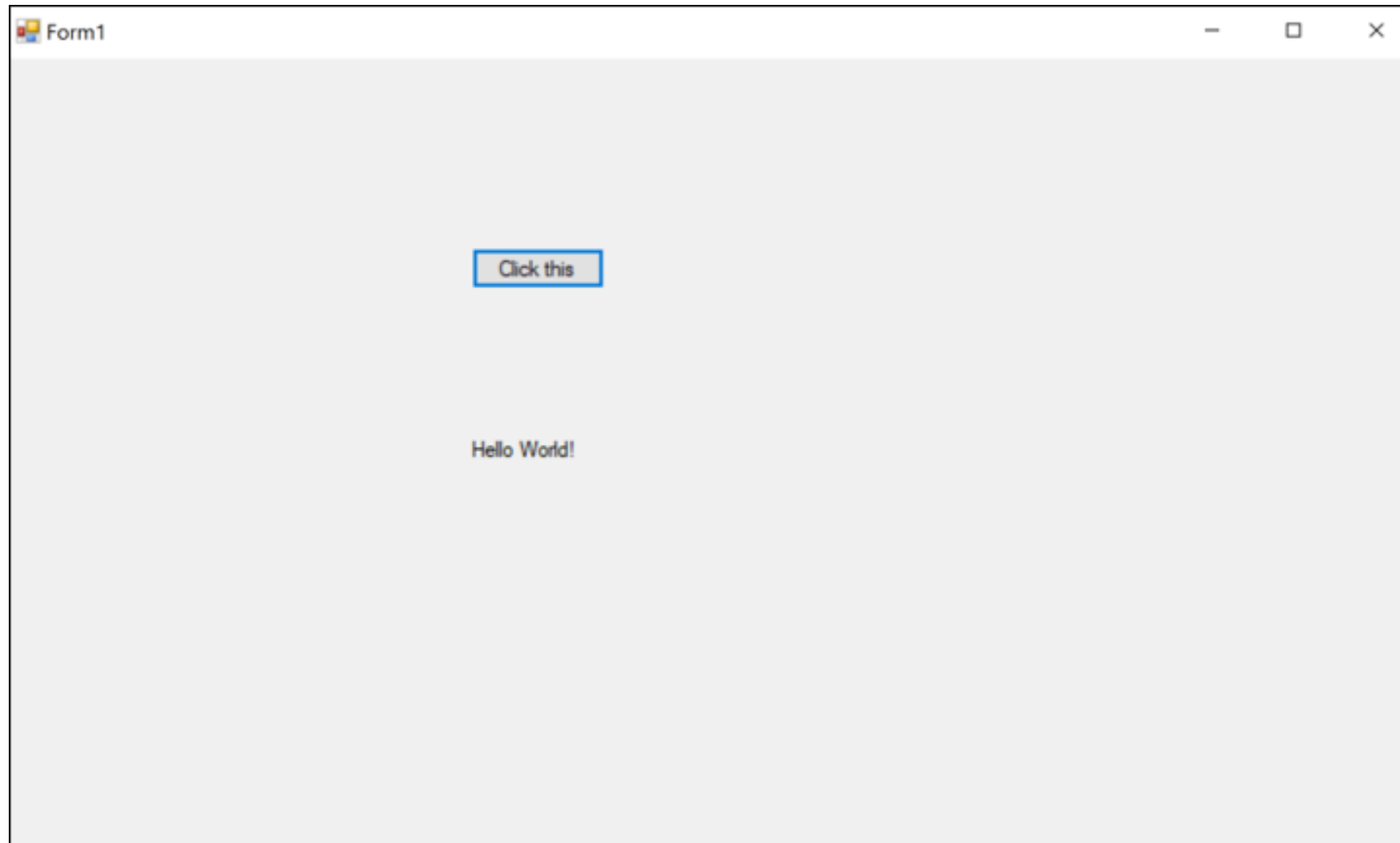
- In the Form1.cs [Design] window, double-click the Click this button to open the Form1.cs window.
- (Alternatively, you can expand Form1.cs in Solution Explorer, and then choose Form1.)
- In the Form1.cs window, after the private void line, type or enter `lblHelloWorld.Text = "Hello World!";`



Run the application

- Select the Start button to run the application.
- Several things will happen. In the Visual Studio IDE, the Diagnostics Tools window will open, and an Output window will open, too. But outside of the IDE, a Form1 dialog box appears. It will include your Click this button and text that says label1.
- Select the Click this button in the Form1 dialog box. Notice that the label1 text changes to Hello World!.

Run the application...



Close the Form1 dialog box to stop running the app.



Thank You