Event Driven Programming

Fundamentals of Event Driven Programming

Chalew Tesfaye
Department of Computer Science
Debre Berhan University
2017

Objectives

- At the end of the lesson students should be able to:
  - ✓ Understand the process of software development
  - ✓ Understand the concept of event driven programming
  - ✓ Understand the .NET framework and its structure
  - ✓ Identify the components of Visual Studio IDE
  - ✓ Build C# project using Visual Studio IDE
  - ✓ Debug C# Project

Contents

Event Driven Programming

- Event
  - ✓ things that happen
  - ✓ For example,
    - you click a button, and the program does something interesting.
- Event driven programming
  - ✓ is a programming paradigm in which the flow of program execution is determined by *events*
  - ✓ For example a user action such as a mouse click, key press, or a message form the operating system or another program
- An event-driven application is designed to
  - ✓ detect events as they occur, and then
  - ✓ deal with them using an appropriate *event-handling procedure*

Event …

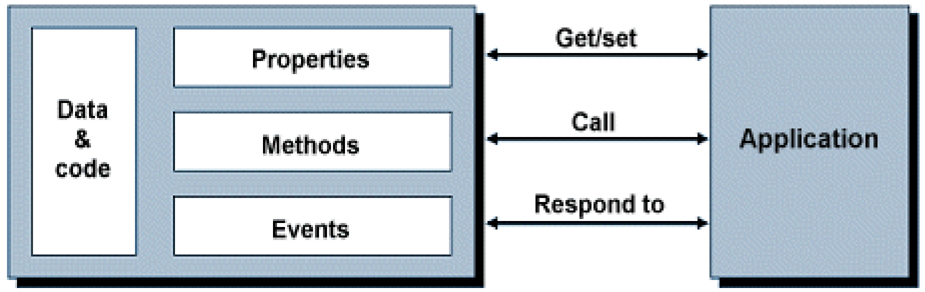
- Event driven, code-behind programming…
  - ✓ Event source
  - ✓ Event
    - individual user actions are translated into “events”
  - ✓ Event listener
  - ✓ Event handler
    - events are passed, 1 by 1, to application for processing
- A visual programming IDE such as MS Visual Studio
  - ✓ provides much of the code for detecting events and
  - ✓ Wires event listener to event handler automatically when a new application created
- The programmer concentrates on
  - ✓ User Interface design
  - ✓ Writing the event handler/the code

Event …

- Structural Development >
  - ✓ Procedural Programming
    - Event handlers implemented as subroutines
    - The flow of program execution was
      - determined by the programmer, and
      - controlled from within the application’s main routine.
- Modern event-driven program,
  - ✓ there is no visible flow of control
  - ✓ The main routine is an event-loop that
    - waits for an event to occur, and then
    - invokes the appropriate event-handling routine
  - ✓ Since the code for this event loop is
    - usually provided by the event-driven development environment or framework
    - largely invisible to the programmer, application is a collection of event handler routines
  - ✓ Accelerated by the introduction of GUI

Event ...

- OO Development >
  - ✓ OO Programming
    - > Object
      - Real world entity
    - > Class
      - is an object template
      - defines the
        - attributes,
        - methods and
        - events that will be implemented in any object created with it.
    - > An *object* is an *instance* of a class



The relationship between an object and an application

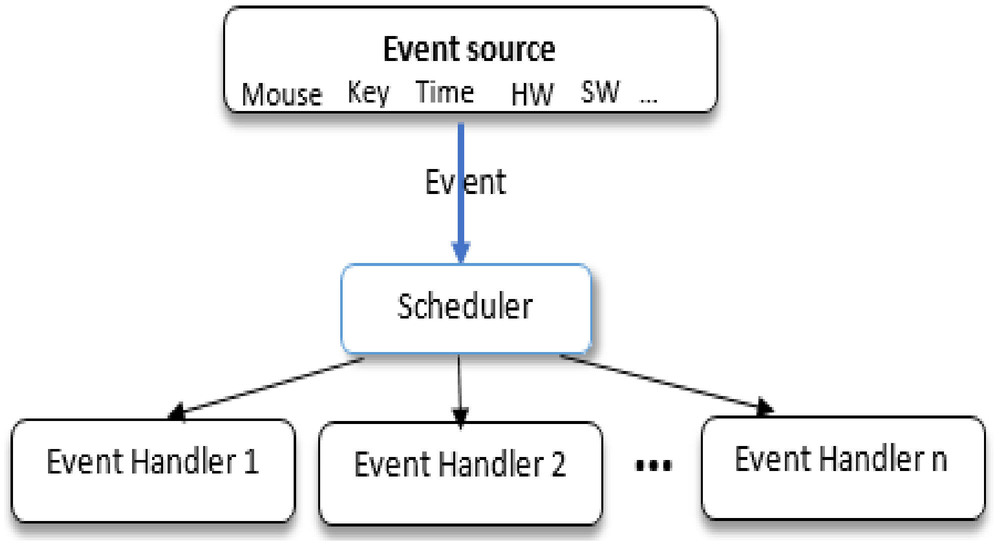
Credit: www.technologyuk.net

How event-driven programming works?

- The central element of an event-driven application is
  - ✓ a scheduler/listener
    - > that receives a stream of events and
    - > passes each event to the relevant event-handler
  - ✓ continue to remain active until it encounters an event
    - > e.g. "End_Program“ - that causes it to terminate the application
- Event includes
  - ✓ actions performed by the user during the execution of a program
    - > Click event, Key press event, Speak event, Touch event, …
  - ✓ messages generated by the operating system or another application
  - ✓ an interrupt generated by a peripheral device or system hardware
  - ✓ Timely generated message
  - ✓ Message sent via network from other system
- The events are dealt with by a central event-handler (usually called a *dispatcher* or *scheduler or listener*)
  - ✓ that runs continuously in the background and waits for an even to occur.
- When an event *does* occur,
  - ✓ the scheduler/ listener must determine the type of event and call the appropriate event-handler to deal with it.
- The information passed to the event handler by the scheduler will vary, but will include sufficient information to allow the event-handler to take any action necessarily.

How event …

```
do forever: // the main scheduler loop

 get event from input stream

 if event type == EndProgram:
   quit // break out of event loop

 else if event type == event_01:
   call event-handler for event_01 with event parameters

 else if event type == event_02:
   call event-handler for event_02 with event parameters

 .
 .
 .

 else if event type == event_nn:
   call event-handler for event_nn with event parameters

 else handle unrecognized event // ignore or raise exception

end loop
```



A simple event-driven programming paradigm

Credit: www.technologyuk.net

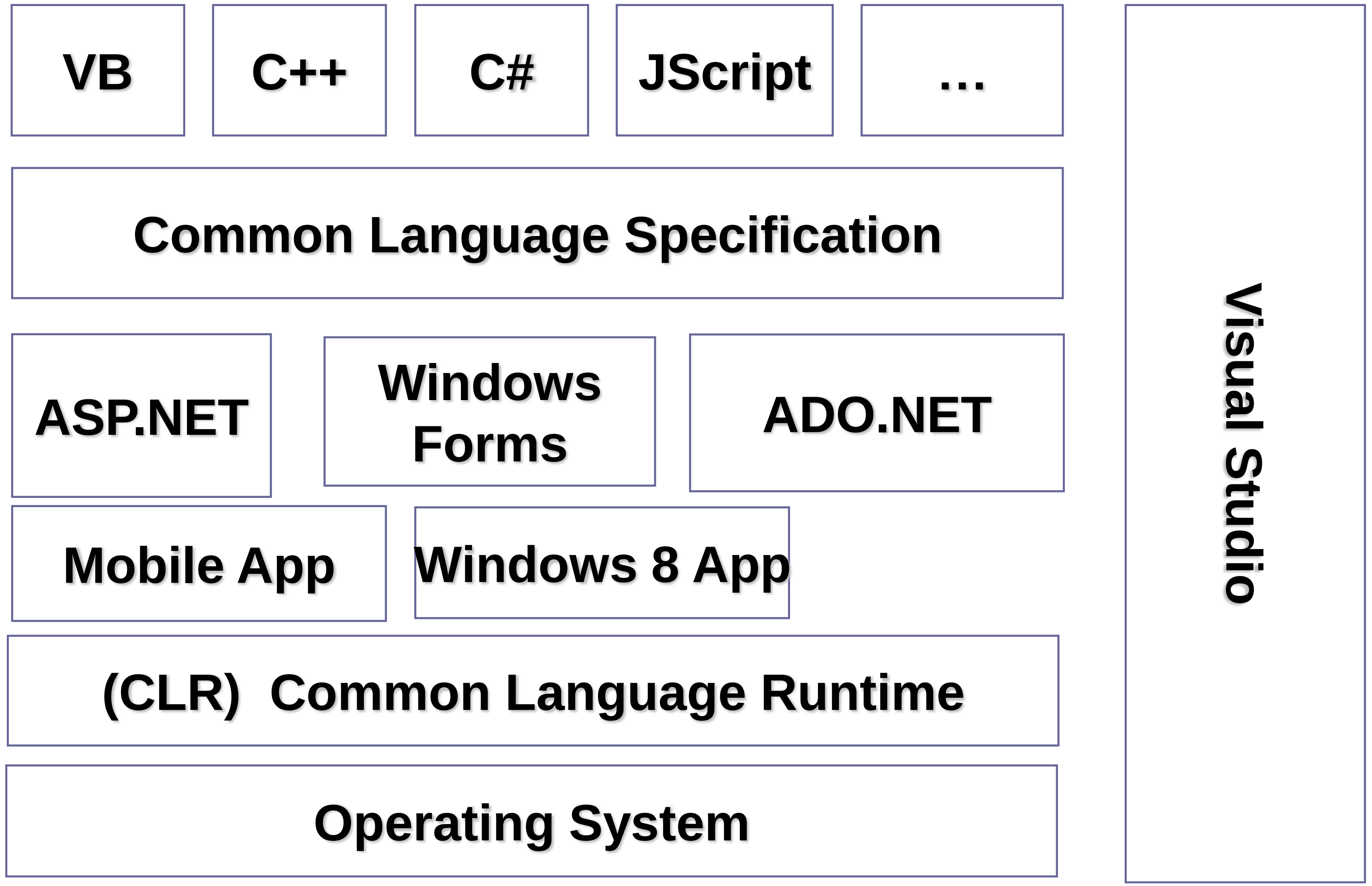
Event-driven programming Language

- Almost all object-oriented and visual languages support event-driven programming
  - ✓ C#, Visual Basic, Visual C++ and Java are examples of such languages.
- A visual programming IDE such as Visual Studio
  - ✓ provides much of the code for detecting events automatically when a new application is created
- In this course we use C#, to develop event driven program
- C#
  - ✓ pronounced as see sharp
  - ✓ is a multi-paradigm programming language
  - ✓ enables developers to build a variety of secure and robust applications that run on the .NET Framework
  - ✓ Can used to create
    - Windows client applications,
    - Windows Store apps
    - XML Web services,
    - distributed components,
    - client-server applications,
    - database applications, and much, much more.

Introduction to .NET framework

- .NET is
  - ✓ pronounced as “dot net”
  - ✓ a software framework that runs primarily on Microsoft Windows.
  - ✓ provides a common platform to Execute or, Run the applications developed in various programming languages.
  - ✓ Microsoft announced the .NET initiative in July 2000.
  - ✓ The main intention was to bridge the gap in **interoperability** between services of various programming languages
- The .NET Framework is designed to fulfill the following objectives:
  - ✓ Provide object-oriented programming environment
  - ✓ Provide environment for developing various types of applications, such as Windows-based applications, Web-based applications, Windows 8 app, Mobile app
  - ✓ To ensure that code based on the .NET Framework can integrate with any other code

.NET Framework …



.NET Framework …

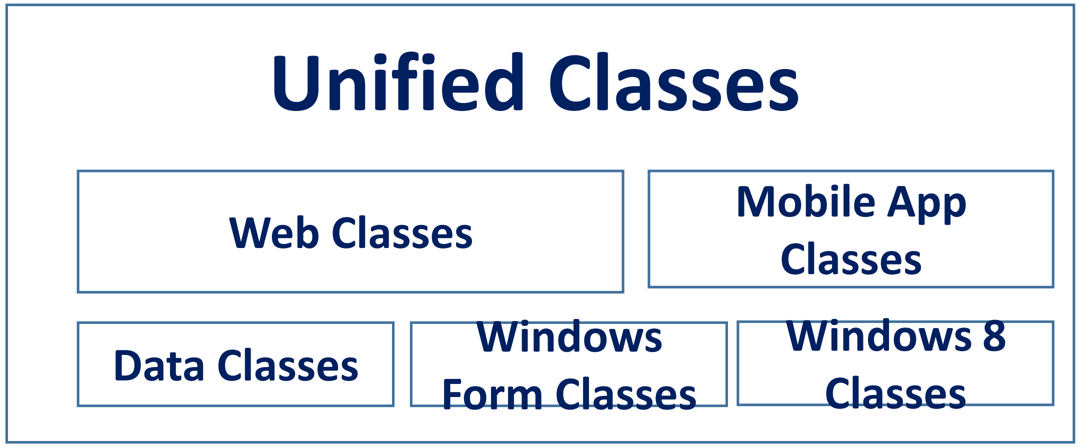
- The .NET Framework consists of:
  - ✓ The Common Language Specification (CLS)
    - > contains guidelines, that language should follow so that they can communicate with other .NET languages
    - > responsible for Type matching.
  - ✓ The Framework Base Class Libraries (BCL)
    - > a consistent, object-oriented library of prepackaged functionality and Applications.
  - ✓ The Common Language Runtime (CLR)
    - > A language-neutral development & execution environment that provides common runtime for application execution .

.NET Framework …

- Common Language Specification (CLS) performs the following functions:
  - ✓ Establishes a framework that helps enable
    - > cross-language integration,
    - > type safety, and
    - > high performance code execution
  - ✓ Provides an object-oriented model
    - > that supports the complete implementation of many programming languages
  - ✓ Defines rules that languages must follow,
    - > which helps ensure that objects written in different languages can interact with each other

.NET Framework …

- .NET Framework Base Class Library (FBC)
  - ✓ The Class Library is a comprehensive, object-oriented collection of reusable types
  - ✓ These class library can be used to develop applications that include:
    - Traditional command-line applications
    - Graphical user interface (GUI) applications
    - Applications based on the latest innovations provided by ASP.NET
      - Web Forms
      - XML Web services
    - Windows 8 Apps
    - Mobile Apps

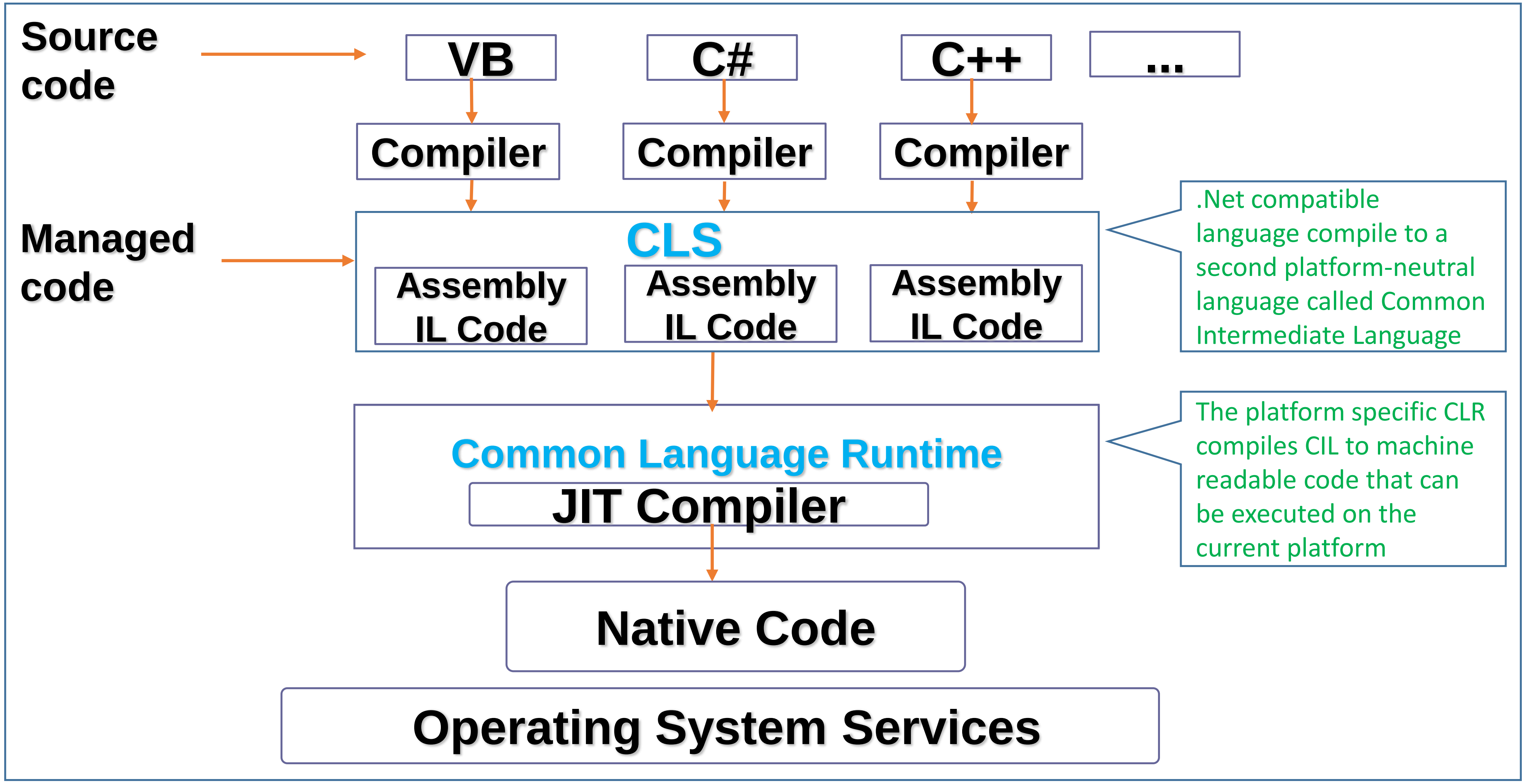


.NET Framework …

- Common Language Runtime (CLR) ensures:
  - A common runtime environment for all .NET languages
  - Uses Common Type System (strict-type & code-verification)
  - Memory allocation and garbage collection
  - Intermediate Language (IL) to native code compiler. Which Compiles MSIL code into native executable code
  - Security and interoperability of the code with other languages
- Over 36 languages supported today
  - C#, VB, Jscript, Visual C++ from Microsoft
  - Perl, Python, Smalltalk, Cobol, Haskell, Mercury, Eiffel, Oberon, Oz, Pascal, APL, CAML, Scheme, etc.

.NET Framework …

Execution in CLR

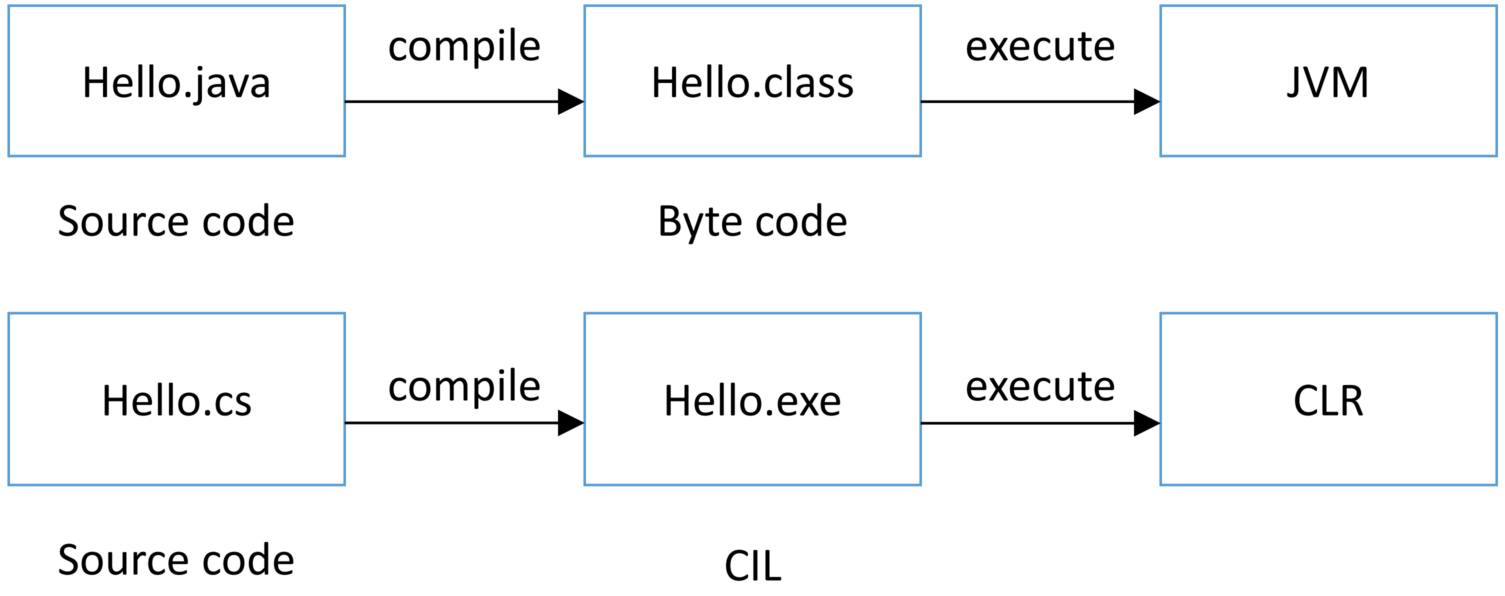


.Net compatible language compile to a second platform-neutral language called Common Intermediate Language

The platform specific CLR compiles CIL to machine readable code that can be executed on the current platform

.NET Framework …

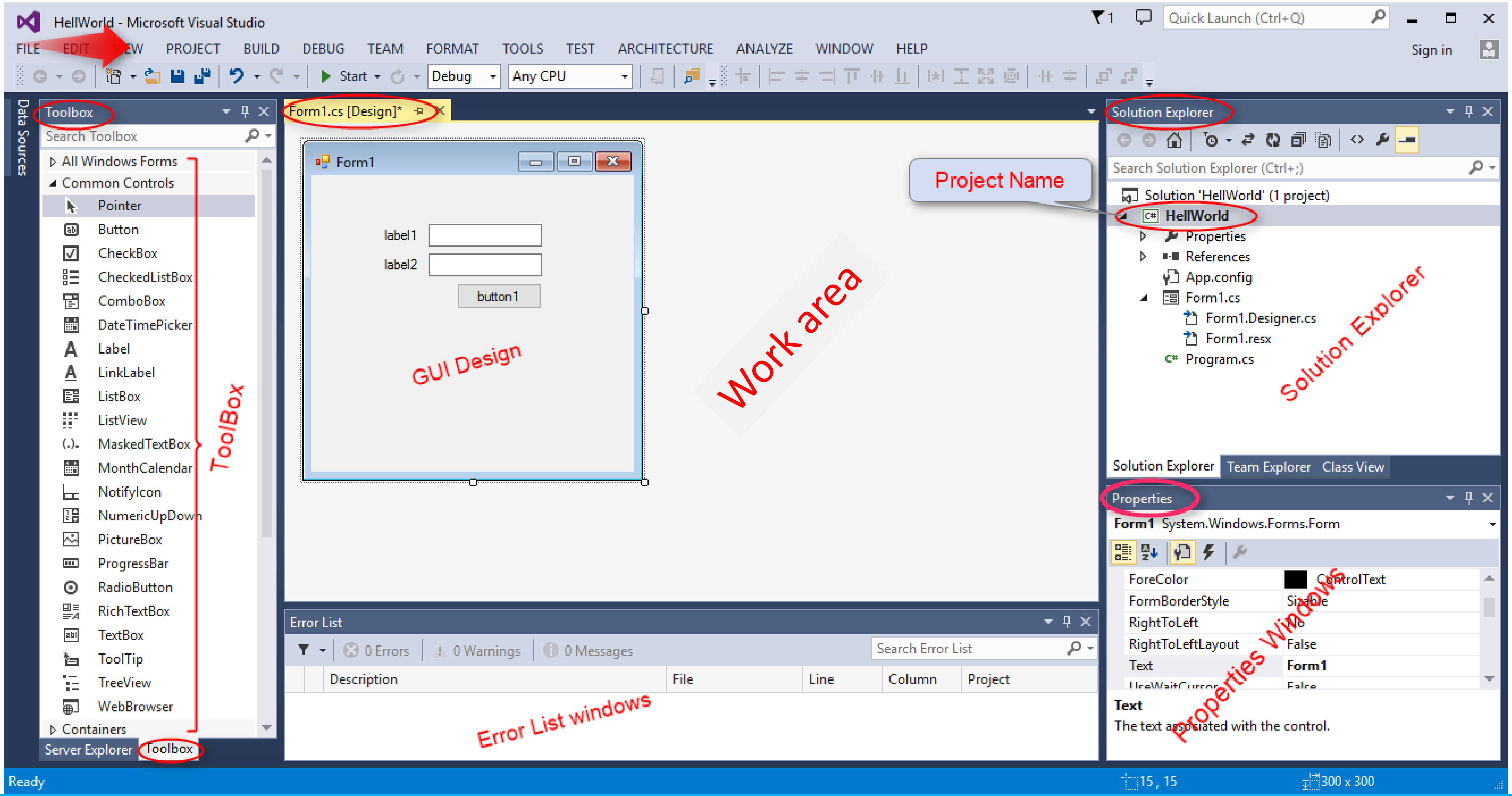
Comparison to Java



Visual Studio IDE

- Microsoft has introduced Visual Studio IDE
  - ✓ which is a set of tool in a single application for developing .NET applications by using
    - programming languages such as VB, C#, VC++ and VJ#, etc.
  - ✓ used to create applications such as Windows Store apps, Windows desktop apps, console apps, Web apps, and Windows Phone apps
- Visual Studio provides
  - ✓ Automatic skeleton code generation
  - ✓ Rapid coding experience
  - ✓ Everything at your fingertips
    - Different tools, easily navigability
  - ✓ Customizability and
  - ✓ Extensibility

Visual Studio …

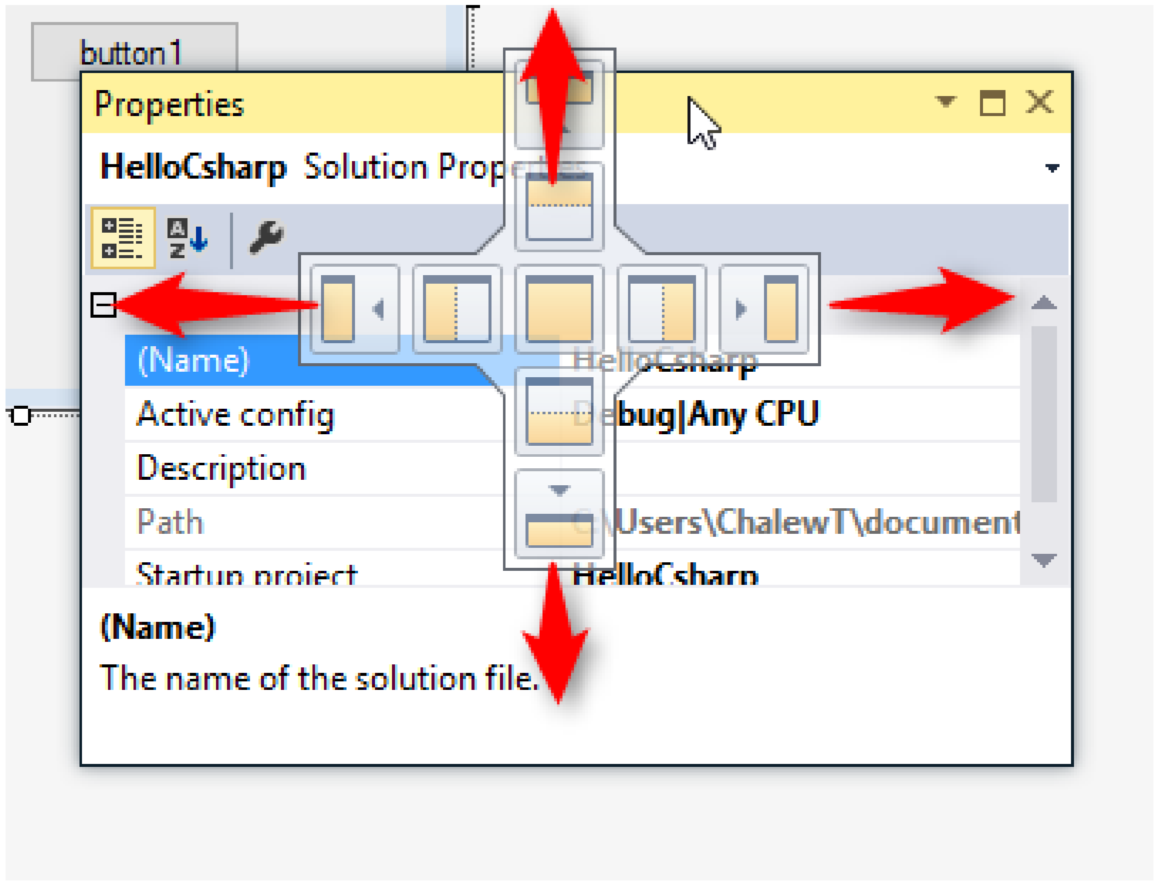


Navigating the Visual Studio Environment

- The Menu
  - ✓ The menu bar is a standard part of most windows applications.
  - ✓ “File,” “Edit,” “View,” “Tools,” and so on.
- Toolbar
  - ✓ contains frequently accessed functionality that is a subset of what is available via menus
- Work area
  - ✓ use to write code and work with visual designers
- Toolbox
  - ✓ contains a context sensitive list of controls that can be dragged and dropped onto the current designer surface
- Solution Explorer
  - ✓ is where your solutions, projects, and project items will appear
- Status Bar
  - ✓ communicates what is happening with VS at the current time
- Much more

Managing Visual Studio Windows

- Expanding and Collapsing Windows
- Docking Windows
- Floating Windows
- Tabbed Windows
- Closing and Opening Windows
- Resetting All Settings

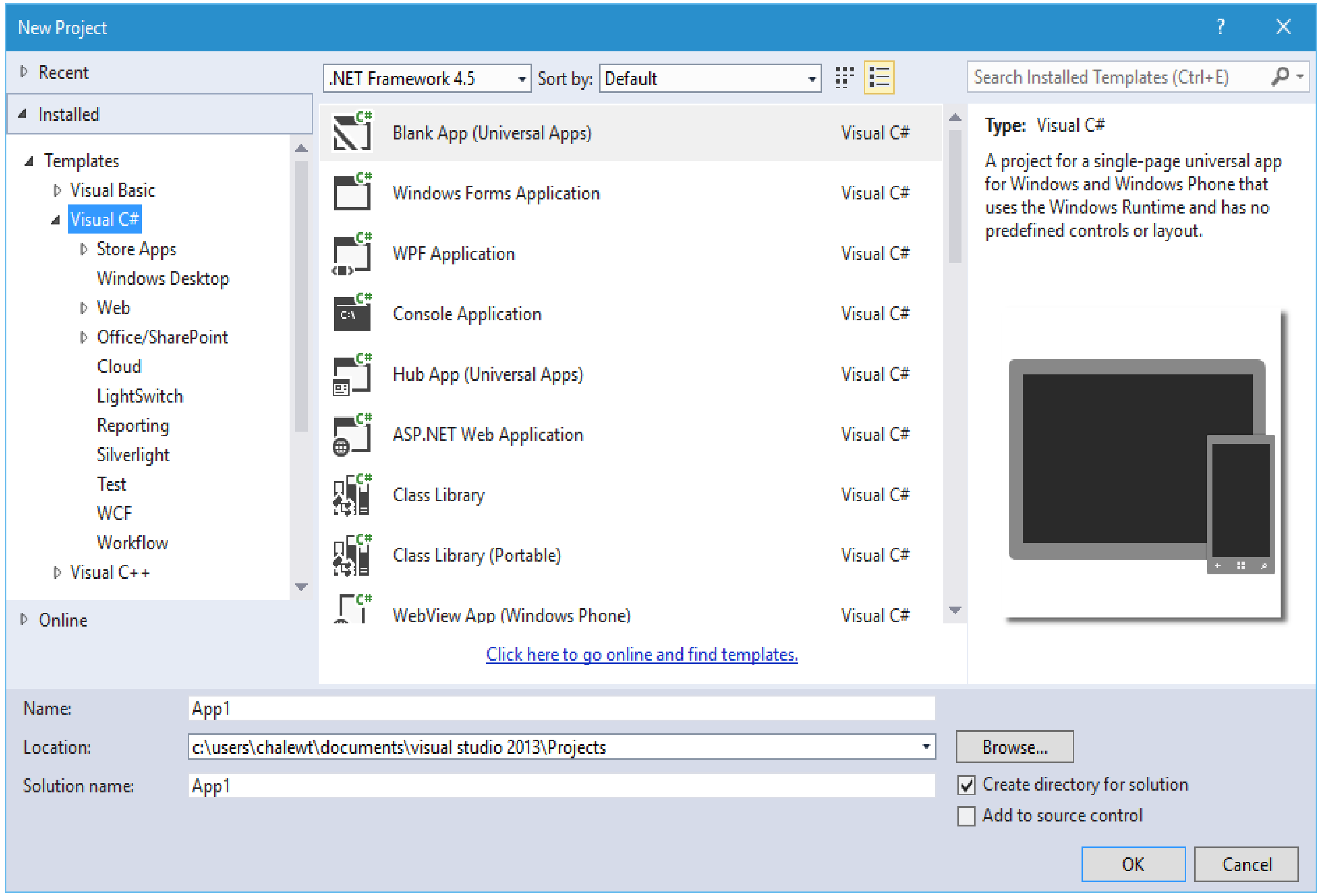


(Name)	HelloCsharp
Active config	Debug|Any CPU
Description	
Path	C:\Users\ChalewT\document
Startup project	HelloCsharp

(Name)
The name of the solution file.

Familiarization with Visual Studio Project Types

- Windows Projects
- Store Apps
- Web Projects
- Office Projects
- SharePoint Projects
- Database Projects



Blank App (Universal Apps)	Visual C#
Windows Forms Application	Visual C#
WPF Application	Visual C#
Console Application	Visual C#
Hub App (Universal Apps)	Visual C#
ASP.NET Web Application	Visual C#
Class Library	Visual C#
Class Library (Portable)	Visual C#
WebView App (Windows Phone)	Visual C#

Type: Visual C#

A project for a single-page universal app for Windows and Windows Phone that uses the Windows Runtime and has no predefined controls or layout.

Name: App1

Location: c:\users\chalewt\documents\visual studio 2013\Projects

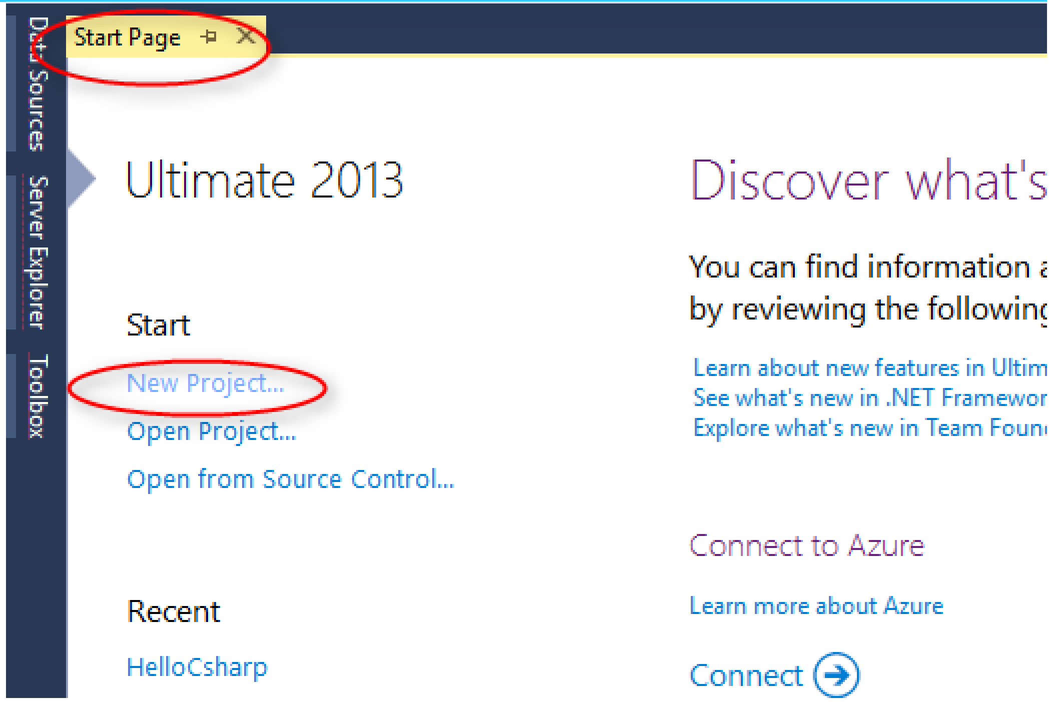
Solution name: App1

[✓] Create directory for solution

[ ] Add to source control

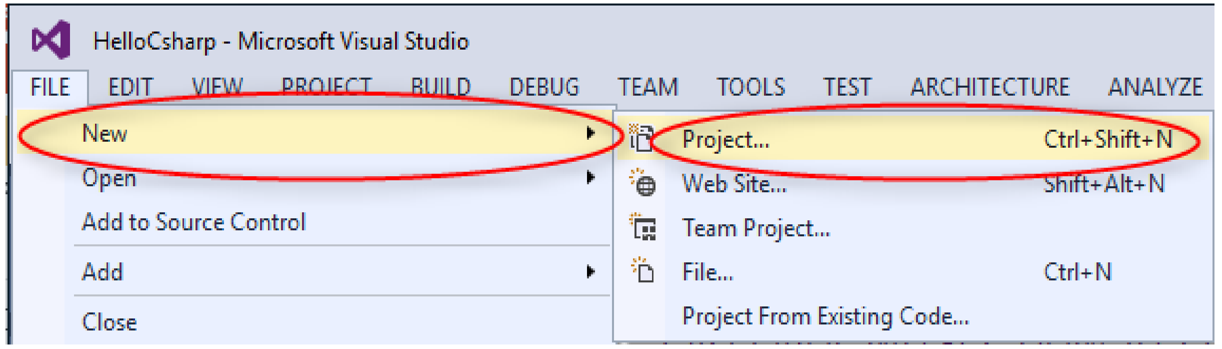
Building Your First Application in VS 2013

- Start Visual Studio 2013
- New Project
  - ✓ Select project template
  - ✓ Select project
- The most common applications are:
  - ✓ Windows Form Application
  - ✓ Console Application
  - ✓ WPF Application
  - ✓ Windows Store App
  - ✓ ASP.NET Web Application
  - ✓ Silverlight Application
- Provide project name, location

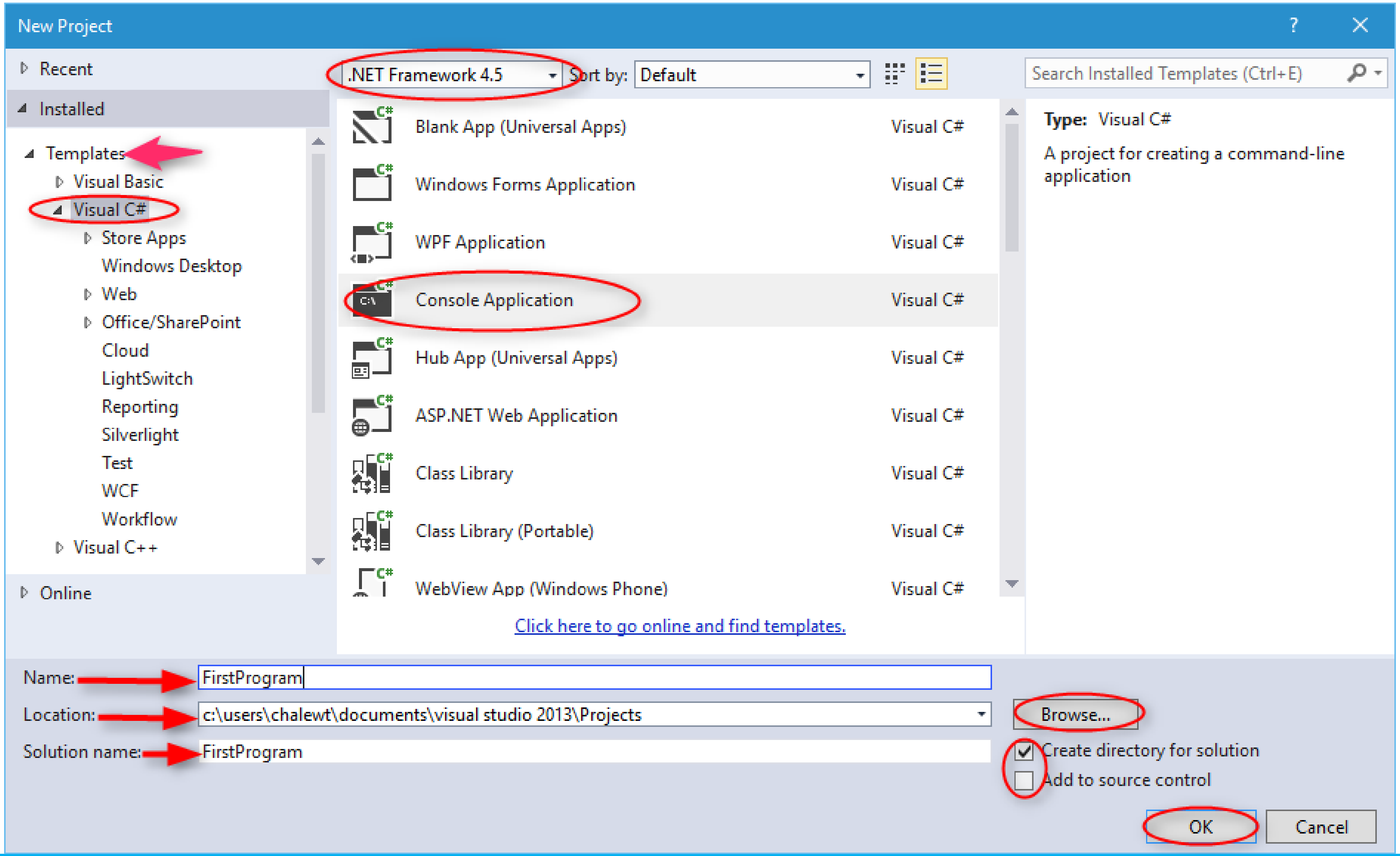


You can find information a
by reviewing the following

Learn about new features in Ultim
See what's new in .NET Framewor
Explore what's new in Team Foun



Building Your First Application …



New Project

- Recent
- Installed
  - Templates
    - Visual Basic
    - Visual C#
      - Store Apps
      - Windows Desktop
      - Web
      - Office/SharePoint
      - Cloud
      - LightSwitch
      - Reporting
      - Silverlight
      - Test
      - WCF
      - Workflow
    - Visual C++
- Online

.NET Framework 4.5 · Sort by: Default · Search Installed Templates (Ctrl+E)

Blank App (Universal Apps)	Visual C#
Windows Forms Application	Visual C#
WPF Application	Visual C#
Console Application	Visual C#
Hub App (Universal Apps)	Visual C#
ASP.NET Web Application	Visual C#
Class Library	Visual C#
Class Library (Portable)	Visual C#
WebView App (Windows Phone)	Visual C#

Click here to go online and find templates.

Type: Visual C#

A project for creating a command-line application

Name: FirstProgram

Location: c:\users\chalewt\documents\visual studio 2013\Projects

Solution name: FirstProgram

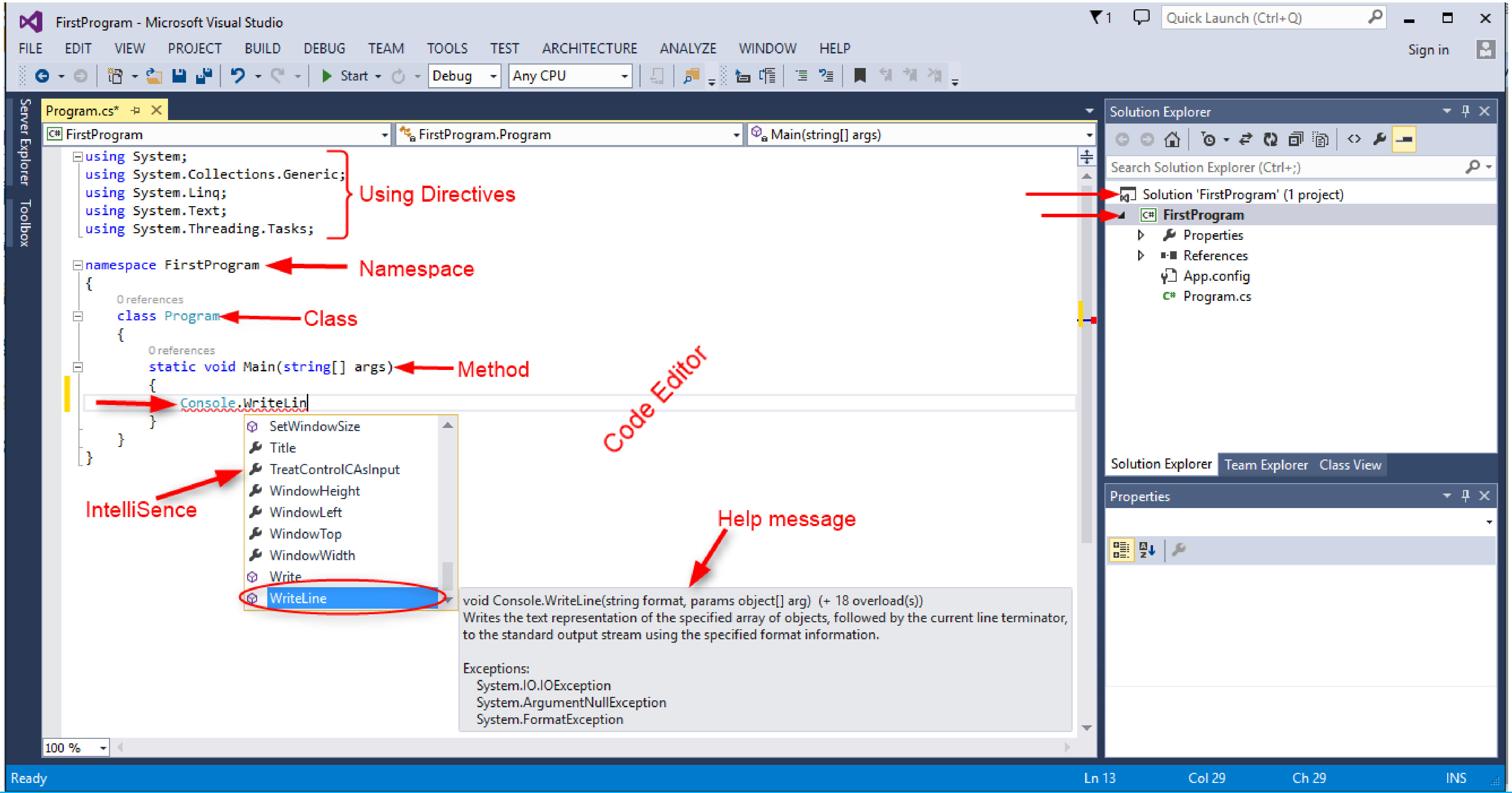
Browse...

[✓] Create directory for solution

[ ] Add to source control

OK · Cancel

Building Your First Application …



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace FirstProgram
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLin
        }
    }
}
```

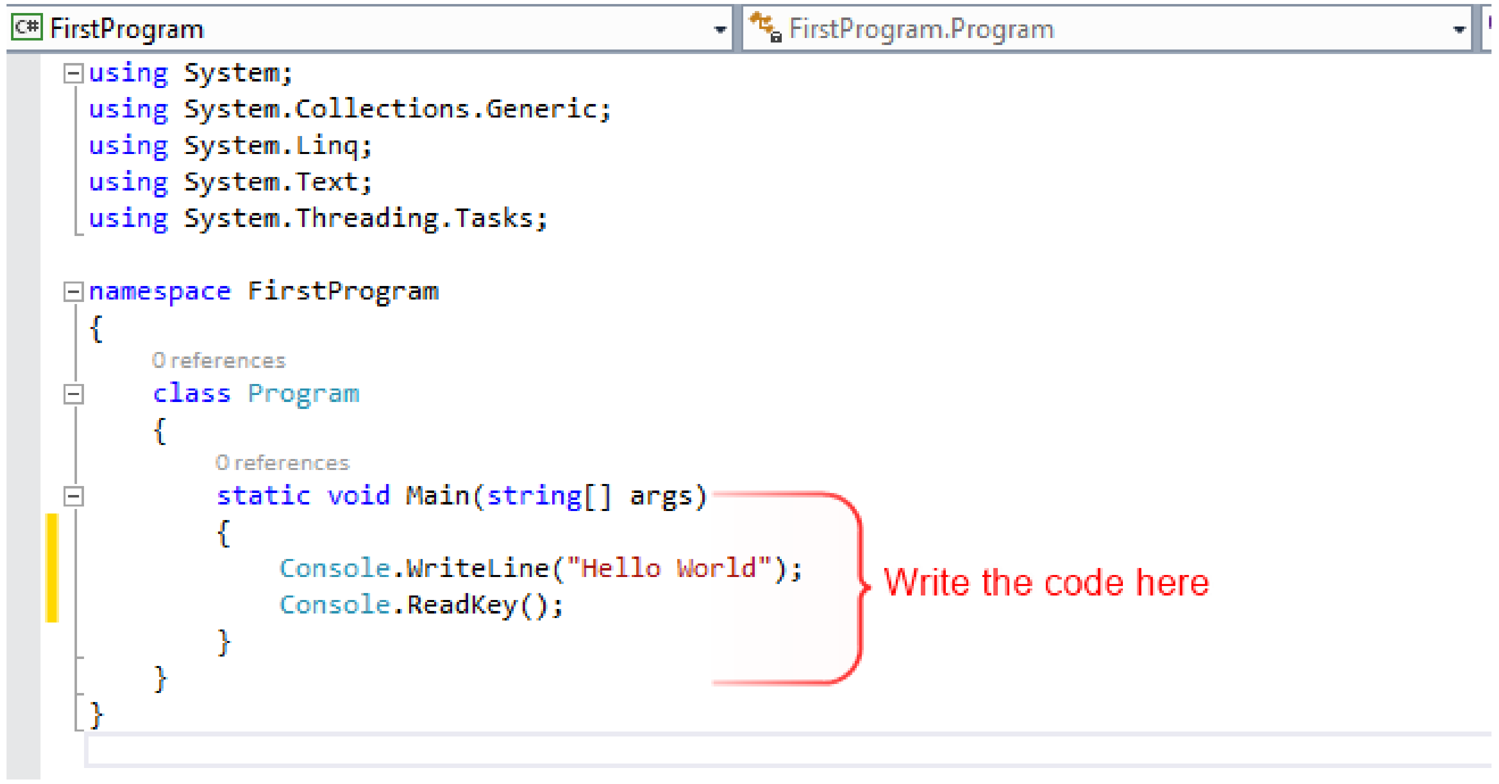
```
SetWindowSize
Title
TreatControlCAsInput
WindowHeight
WindowLeft
WindowTop
WindowWidth
Write
WriteLine
```

void Console.WriteLine(string format, params object[] arg) (+ 18 overload(s))
Writes the text representation of the specified array of objects, followed by the current line terminator, to the standard output stream using the specified format information.

Exceptions:
System.IO.IOException
System.ArgumentNullException
System.FormatException

Building Your First Application ...

- Console Application
  - ✓ Code Editor

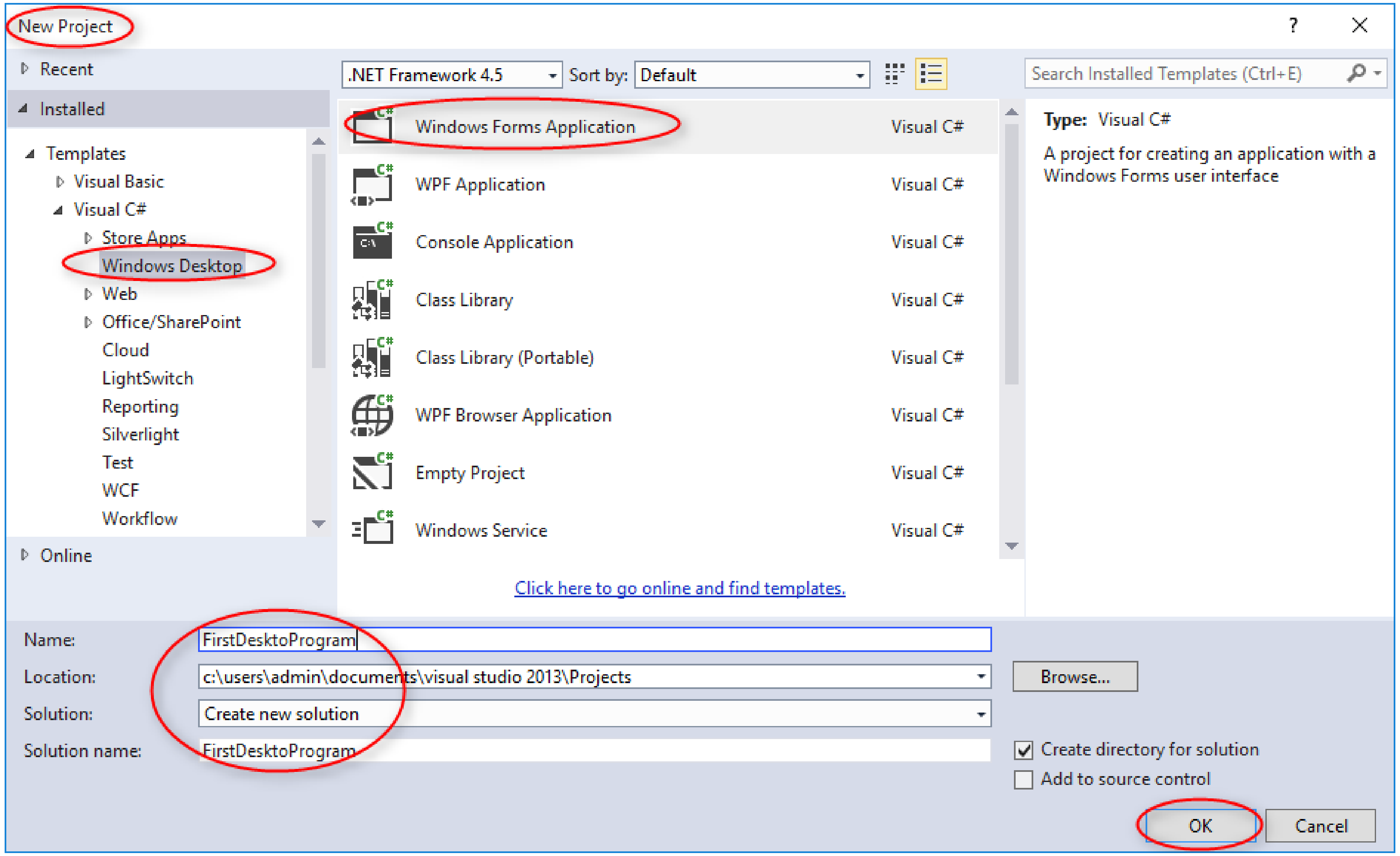


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace FirstProgram
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World");
            Console.ReadKey();
        }
    }
}
```

Write the code here

Building Your First Application …

➤ Desktop Application



Type: Visual C#

A project for creating an application with a Windows Forms user interface

Name: FirstDesktoProgram

Location: c:\users\admin\documents\visual studio 2013\Projects

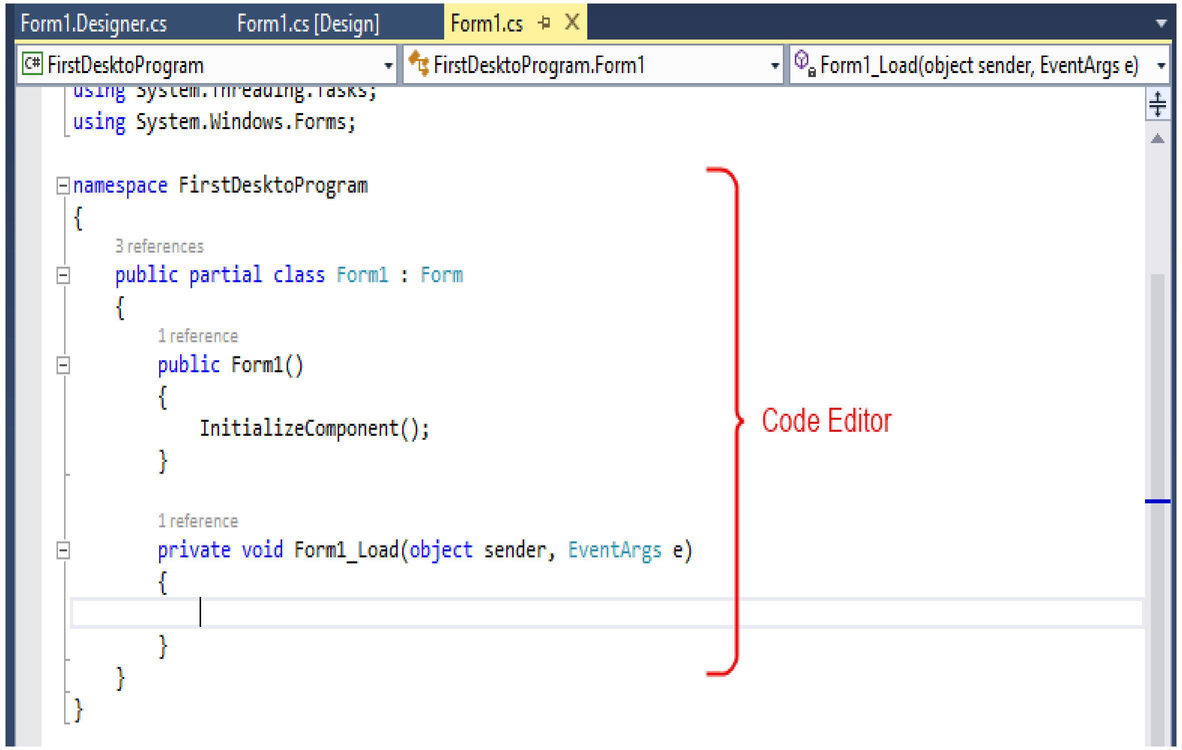
Solution: Create new solution

Solution name: FirstDesktoProgram

☑ Create directory for solution

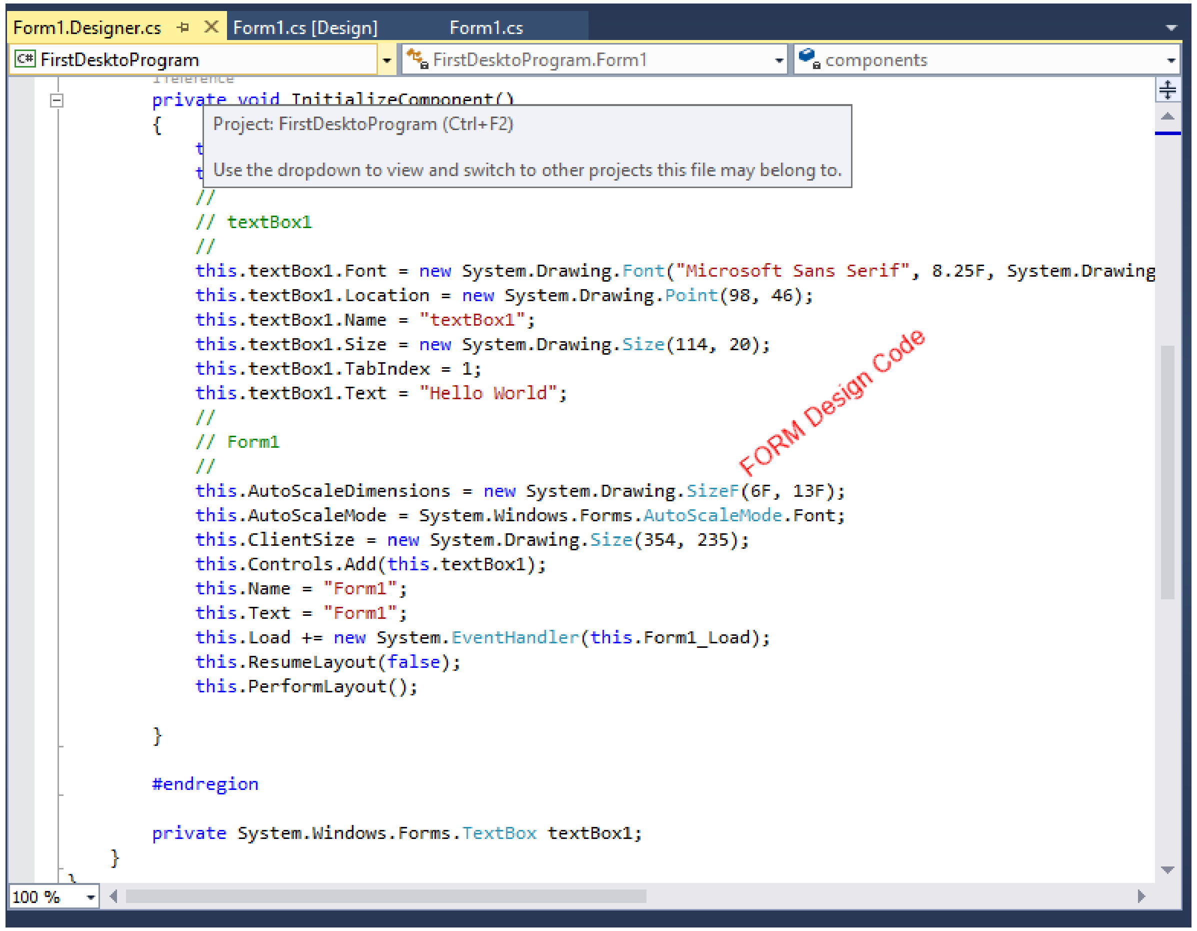
Building Your First Application …

Program Code



```
using System.Threading.Tasks;
using System.Windows.Forms;

namespace FirstDesktoProgram
{
    3 references
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
        }

        1 reference
        private void Form1_Load(object sender, EventArgs e)
        {

        }
    }
}
```

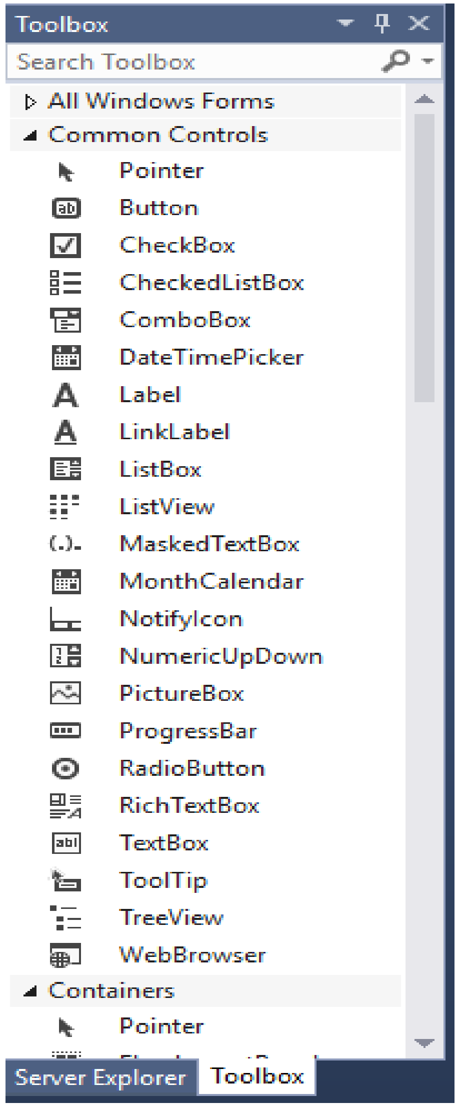
Form Design Code


Form1.Designer.cs
Form1.cs [Design]
Form1.cs
FirstDesktoProgram
FirstDesktoProgram.Form1
components
Project: FirstDesktoProgram (Ctrl+F2)
Use the dropdown to view and switch to other projects this file may belong to.
FORM Design Code
100 %

```
        1 reference
        private void InitializeComponent()
        {
            t
            t
            //
            // textBox1
            //
            this.textBox1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing
            this.textBox1.Location = new System.Drawing.Point(98, 46);
            this.textBox1.Name = "textBox1";
            this.textBox1.Size = new System.Drawing.Size(114, 20);
            this.textBox1.TabIndex = 1;
            this.textBox1.Text = "Hello World";
            //
            // Form1
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(354, 235);
            this.Controls.Add(this.textBox1);
            this.Name = "Form1";
            this.Text = "Form1";
            this.Load += new System.EventHandler(this.Form1_Load);
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.TextBox textBox1;
    }
}
```

Using the intrinsic controls

- Controls
  - ✓ Used to create user interface
  - ✓ TextBox
  - ✓ Label
  - ✓ Button
  - ✓ CheckBox
  - ✓ ListBox
  - ✓ ComboBox
  - ✓ ListView
  - ✓ …


Toolbox
Search Toolbox
All Windows Forms
Common Controls
Pointer
Button
CheckBox
CheckedListBox
ComboBox
DateTimePicker
Label
LinkLabel
ListBox
ListView
MaskedTextBox
MonthCalendar
NotifyIcon
NumericUpDown
PictureBox
ProgressBar
RadioButton
RichTextBox
TextBox
ToolTip
TreeView
WebBrowser
Containers
Pointer
Server Explorer
Toolbox

Methods, Properties, and Events

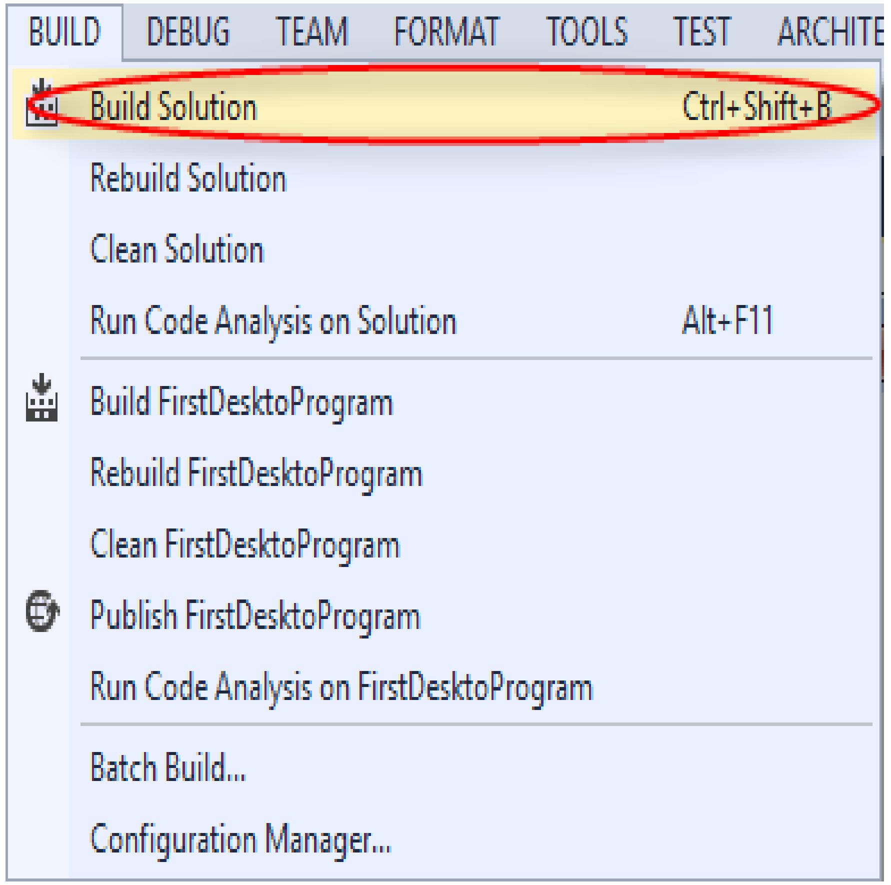
- Methods
  - ✓ Are extremely useful because they allow you to separate your logic into different units
  - ✓ Are similar to functions, procedure or subroutine used in other programming languages
  - ✓ The difference is that a method is always a part of a class
- Properties
  - ✓ Attribute of an object
  - ✓ Example
    - TextBox – Color, Font Family, Font Size
- Events
  - ✓ An action the object responds
  - ✓ Example
    - TextBox – TextChanged Event
    - Button – Click event, …

Debugging the Program

- When our program contains errors, also known as bugs, we must find and remove them,
  - ✓ i.e. we need to debug the program.
- The debugging process includes:
  - ✓ Noticing the problems (bugs);
  - ✓ Finding the code causing the problems;
  - ✓ Fixing the code so that the program works correctly;
  - ✓ Testing to make sure the program works as expected after the changes are made
- The process can be repeated several times until the program starts working correctly
- Visual Studio can help by allowing us to check step by step whether everything is working as planned.
- To stop the execution of the program at designated positions we can place breakpoints.

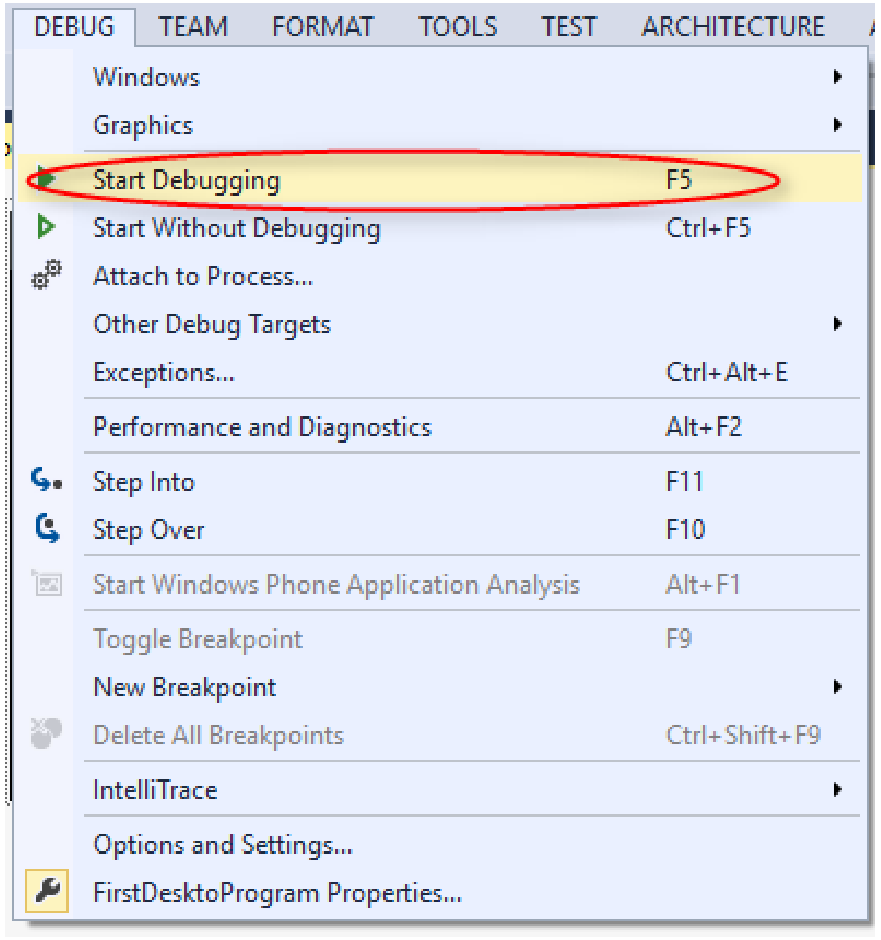
Debugging …

- In Visual Studio we have lots of Build and Debugging Tools
- Build menu
  - ✓ The most used tool is "Build Solution"


BUILD
DEBUG
TEAM
FORMAT
TOOLS
TEST
ARCHITE
Build Solution Ctrl+Shift+B
Rebuild Solution
Clean Solution
Run Code Analysis on Solution Alt+F11
Build FirstDesktoProgram
Rebuild FirstDesktoProgram
Clean FirstDesktoProgram
Publish FirstDesktoProgram
Run Code Analysis on FirstDesktoProgram
Batch Build...
Configuration Manager...

Debugging …

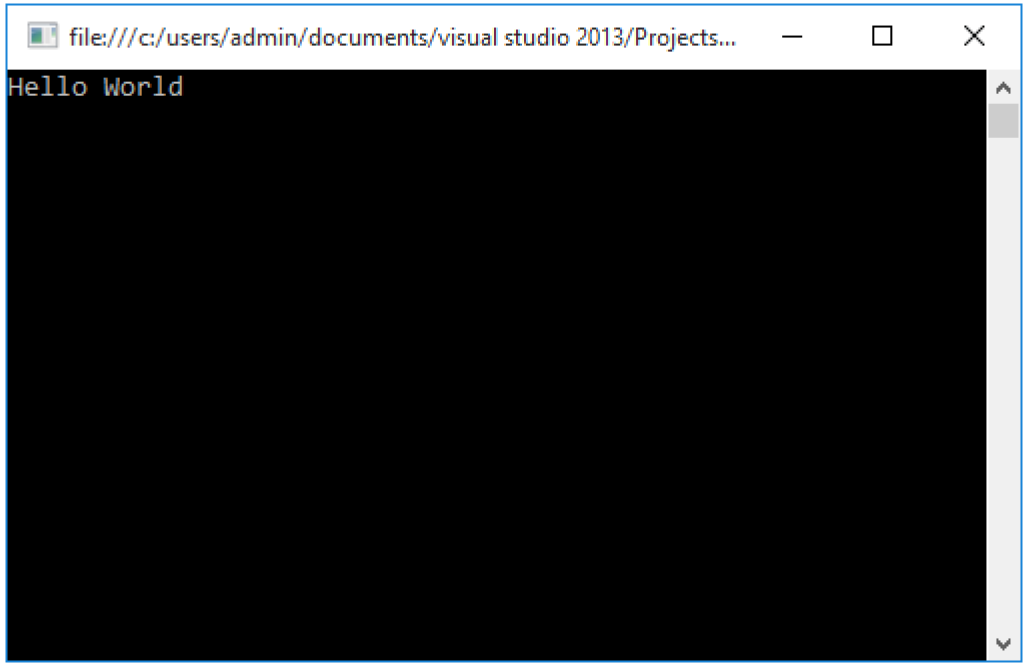
- Debug menu:
  - ✓ The most used tool is “Start Debugging” (Shortcut Key: F5)



DEBUG TEAM FORMAT TOOLS TEST ARCHITECTURE
Windows
Graphics
Start Debugging F5
Start Without Debugging Ctrl+F5
Attach to Process...
Other Debug Targets
Exceptions... Ctrl+Alt+E
Performance and Diagnostics Alt+F2
Step Into F11
Step Over F10
Start Windows Phone Application Analysis Alt+F1
Toggle Breakpoint F9
New Breakpoint
Delete All Breakpoints Ctrl+Shift+F9
IntelliTrace
Options and Settings...
FirstDesktoProgram Properties...

Debugging …

➢ Console Program

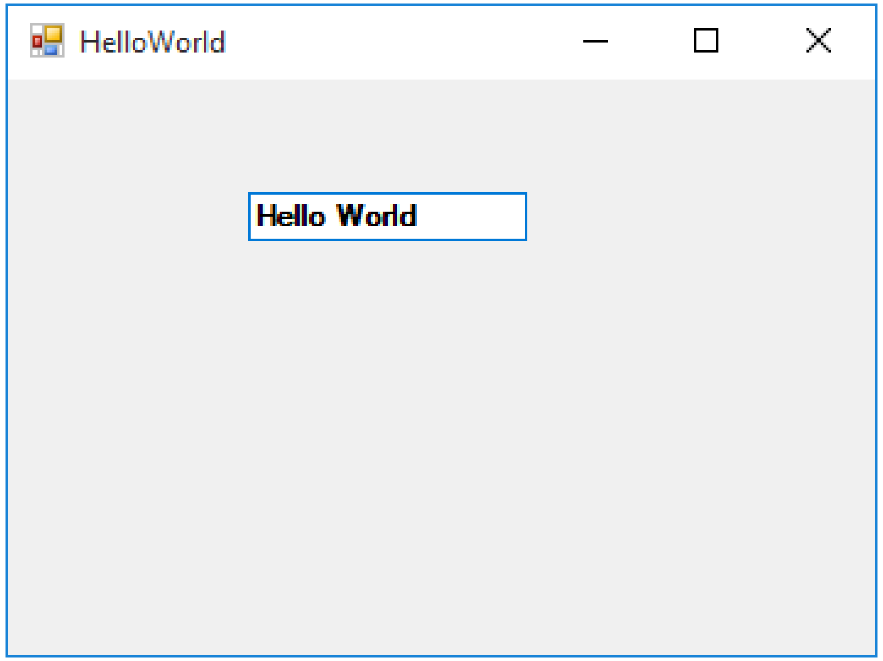


file:///c:/users/admin/documents/visual studio 2013/Projects...

```
Hello World
```

Debugging …

- Desktop program



Hello World

More information on

- John Sharp. Microsoft Visual C# 2013 Step by Step, 2015 Microsoft Press USA
- Joel Murach, Anne Boehm. Murach C# 2012, Mike Murach & Associates Inc USA, 2013
- Lalit Arora. .Net framework with C# programming, 2007 India (HC available in your library)
- http://www.technologyuk.net/computing/software_development/event_driven_programming.shtml, accessed on date 2/18/2016
- Microsoft Corporation. Microsoft Visual Studio 2012 Product Guide, 2012

QUESTIONS

?