Session 1: Getting Started with C#

Objectives

- **Define and describe the .NET Framework**
- **Explain C# language features**
- **Define and describe the Visual Studio 2019 environment**
- **Explain the elements of Microsoft Visual Studio 2019 IDE**
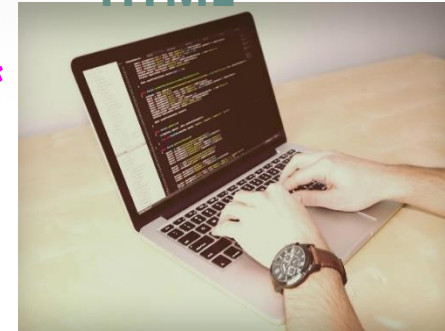
Introduction to .NET Framework 1-2

The .NET Framework:

- Is a revolutionary platform created by Microsoft for application developers

- Includes several tools and technologies that simplify software development

- Can be used to develop different types of applications

- Supports multiple languages and cross-language integration

Introduction to .NET Framework 2-2
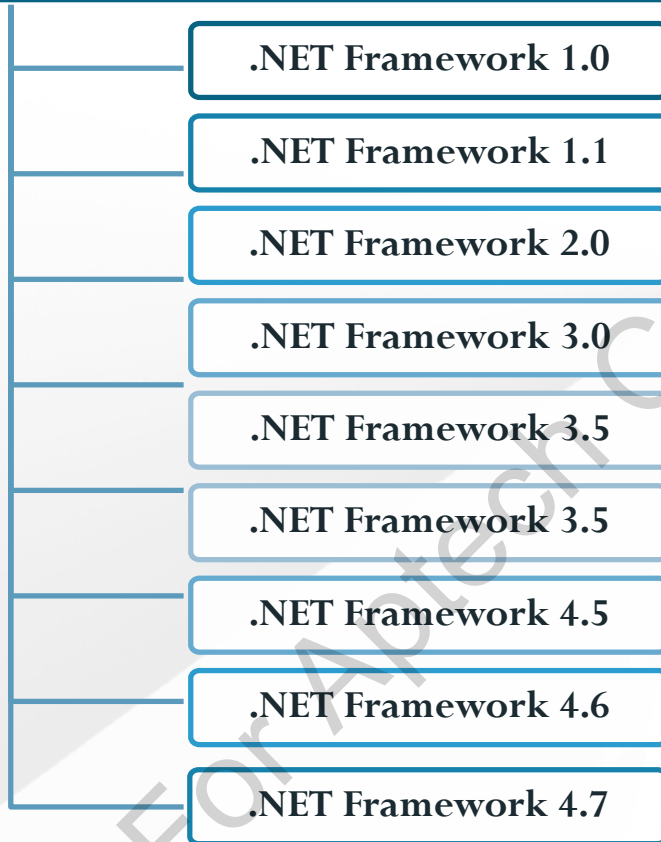
Following are various versions of .NET Framework:

- .NET Framework 1.0
- .NET Framework 1.1
- .NET Framework 2.0
- .NET Framework 3.0
- .NET Framework 3.5
- .NET Framework 3.5
- .NET Framework 4.5
- .NET Framework 4.6
- .NET Framework 4.7

Different Version of .NET Framework 1-3

1.0

Includes CLR, class libraries of .NET Framework and ASP. NET.

1.1

Supported components used to create applications for mobiles.

Supported Oracle databases.

Supported IPv6 protocol and Code Access Security (CAS).

Enabled running assemblies of Windows Forms.

Provided components to create mobile and PDA applications

2.0

Supported for 64-bit hardware platforms

Supported Generic data structures.

Supported for new Web controls

Allowed developers to create graphical devices

Different Version of .NET Framework 2-3

3.0

Windows Presentation Foundation (WPF)

Windows Communication Foundation (WCF)

Windows Workflow Foundation (WF)

Windows CardSpace

3.5

Supported to develop AJAX-enabled Websites

Language Integrated Query (LINQ)

ADO.NET Entity Framework

ADO.NET Data Services

4.0

Dynamic Language Runtime (DLR)

Supported parallel computing that utilizes multi-core capabilities

Provided improvement in ADO.NET, WCF, and WPF

Introduced dynamic dispatch, named parameters, and optional parameters.

Different Version of .NET Framework 3-3

4.5

Enhancement in asynchronous programming

Supported Zip compression

Supported `regex` timeout

Efficient garbage collection

4.8

Continued to be serviced with monthly security and reliability bug fix.

Continued to be included with Windows.

5.0

Created a unified environment and a unified output.

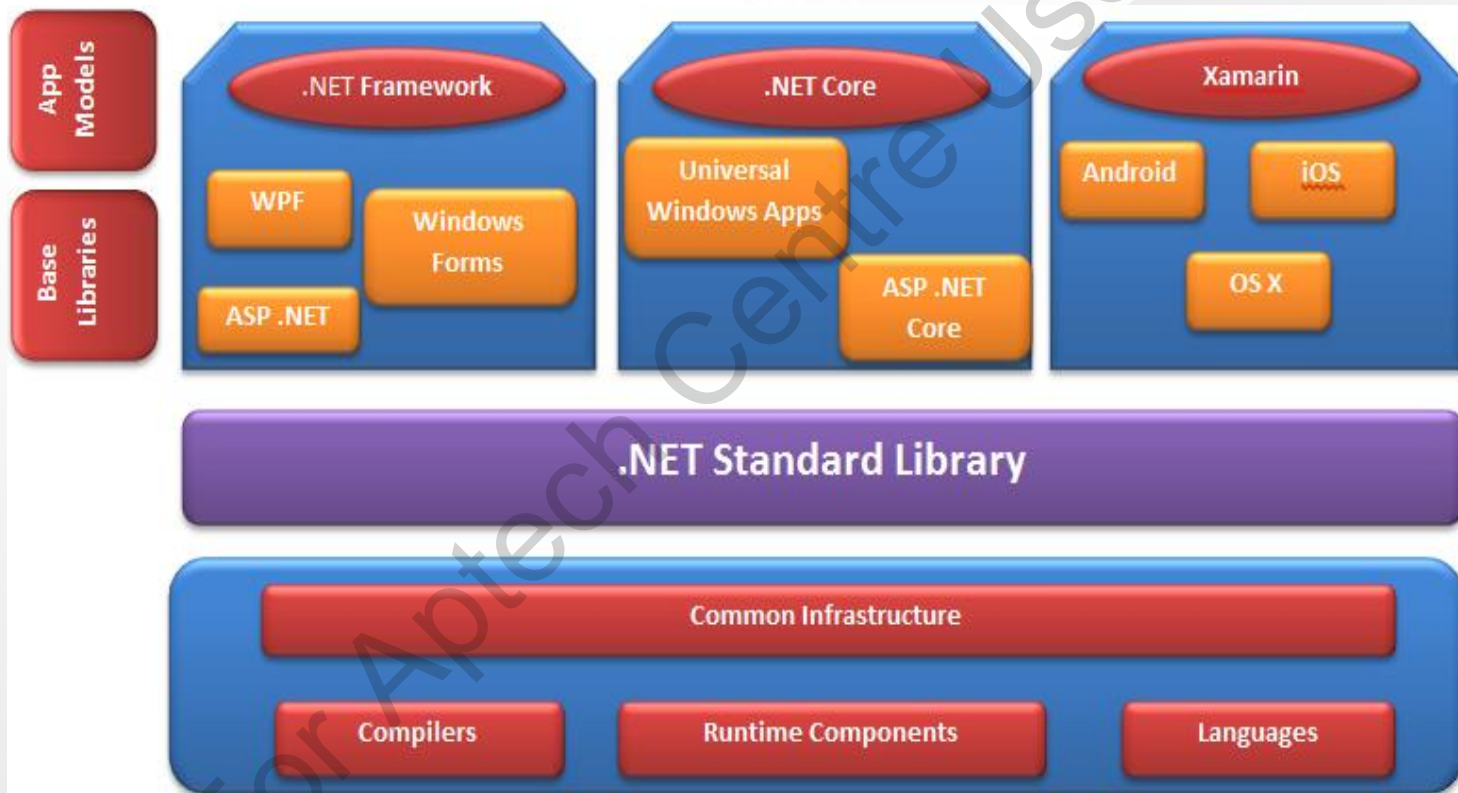Supported all the Visual Studio versions that came after Visual Studio 2017.

Components of the .NET Framework 1-2

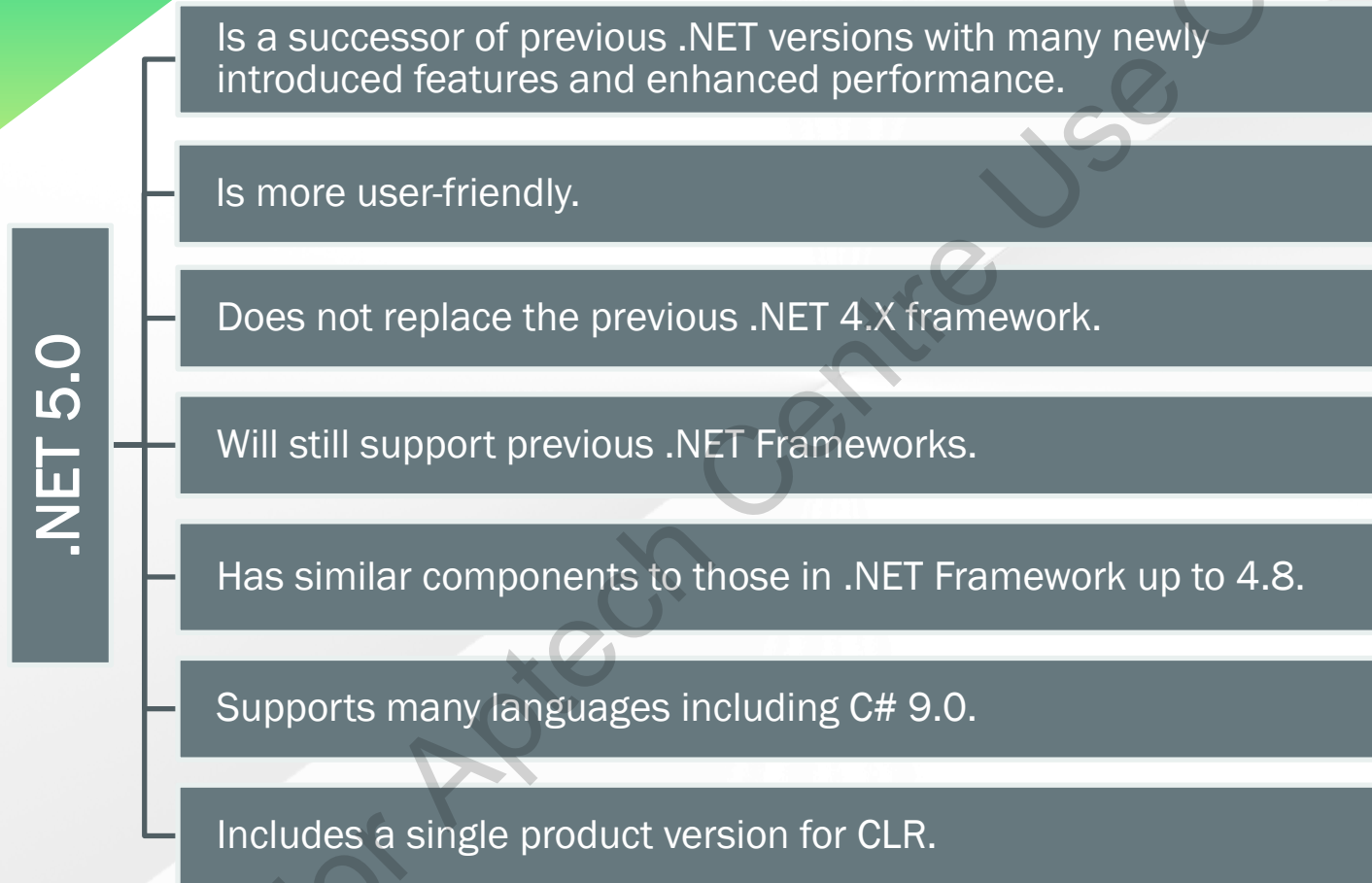| Component	Description
Common Intermediate Language (CIL)	When code written in a .NET compatible language is compiled, output code is in the form of MSIL code (now called as CIL).
Common Language Runtime (CLR)	Backbone of .NET Framework which manages execution of applications and acts as an execution engine.
Dynamic Language Runtime (DLR)	Run-time environment built on top of the CLR to enable interoperability of dynamic languages such as Ruby and Python with the .NET Framework.
FCL	Used to develop applications ranging from traditional command-line to Graphical User Interface (GUI) applications that can be used on the Web.
Common Type System (CTS)	Provides a common way to describe all supported types and establishes a framework for cross-language execution.
Common Language Specification (CLS)	Set of rules that any .NET language should follow to create applications that are interoperable with other languages.
Standard system services	The .NET Framework supports standard system services, such as ADO .NET and XML.

Components of the .NET Framework 2-2

Tiered architecture consisting of several components at each tier.

Two core components include CLR and FCL.

Components of .NET 5.0

.NET 5.0

- Is a successor of previous .NET versions with many newly introduced features and enhanced performance.

- Is more user-friendly.

- Does not replace the previous .NET 4.X framework.

- Will still support previous .NET Frameworks.

- Has similar components to those in .NET Framework up to 4.8.

- Supports many languages including C# 9.0.

- Includes a single product version for CLR.

.NET 5.0 Features

Unified Platform and Updated Target Framework Moniker (TFM)

.NET 5 offers a common set of APIs.

Set of APIs is identified by the `net5.0` TFM.

Before .NET 5, every .NET run-time had its own set of TFMs

Support for single-file applications

.NET 5 provides support for single-file applications.

You do not even require to install the .NET run-time on the target machine.

Language Updates

.NET 5.0 supports the latest version of C# that is C# 9.0 and F# which is F# 5.0.

Compiler feature also enhances C# by adding meta-programming support.

C# source generators also have access to application code.

.NET Multi-platform App UI (MAUI)

.NET 5.0 lays the foundation for .NET MUAI framework.

Developers will be able to build user interfaces for Android, iOS, macOS, and Windows with a single project.

Introduction to C#

C#:

▷ Is an object-oriented language derived from C and C++.

▷ Provides a simple and productive way for developers to create applications.

C# Goals

- Creates a very simple and yet powerful tool for building interoperable, scalable, and robust applications

- Creates a complete object-oriented architecture

- Supports powerful component-oriented development

- Allows access to many features previously available only in C++ while retaining the ease-of-use of a rapid application development tool such as Visual Basic

- Provides familiarity to programmers coming from C or C++ background

- Allows to write applications that target both desktop and mobile devices

Language Features 1-2

Object-oriented Programming	• C# application programming focuses on objects so that code written once can be reused. This helps reduce time and effort on the part of developers.
Type-safety Checking	• Uninitialized variables cannot be used in C#. Overflow of types can be checked. C# is a case-sensitive language.
Garbage Collection	• Performs automatic memory management from time to time and spares the programmer the task.
Standardization by European Computer Manufacturers Association (ECMA)	• This standard specifies the syntax and constraints used to create standard C# programs.
Generic Types and Methods	• Generics are a type of data structure that contains code that remains the same throughout but the data type of the parameters can change with each use.
Iterators:	• Enable looping (or iterations) on user-defined data types with the foreach loop.
Static Classes:	• Contain only static members and do not require instantiation.
Partial Classes:	• Allow the user to split a single class into multiple source code (.cs) files.

Language Features 2-2

Anonymous Methods:	• Enable the user to specify a small block of code within the delegate declaration.
Methods with Named Arguments:	• Enable the user to associate a method argument with a name rather than its position in the argument list.
Methods with Optional Arguments:	• Allow the user to define a method with an optional argument with a default value. The caller of the method may or may not pass the optional argument value during the method invocation.
Nullable Types:	• Allow a variable to contain a value that is undefined.
Accessor Accessibility:	• Allows the user to specify the accessibility levels of the get and set accessors.
Auto-implemented Properties:	• Allow the user to create a property without explicitly providing the methods to get and set the value of the property.
Parallel Computing:	• In .NET Framework and C#, there is strong support for parallel programming using which develop efficient, fine-grained, and scalable parallel code without working directly with threads or the thread pool.

Applications of C#

Some applications of C# are as follows:

- Web applications
- Web services
- Gaming applications
- Large-scale enterprise applications
- Simple standalone desktop applications such as Library Management Systems, Student Mark Sheet generation, and so on
- Complex distributed applications that can spread over a number of cities or countries
- Cloud applications
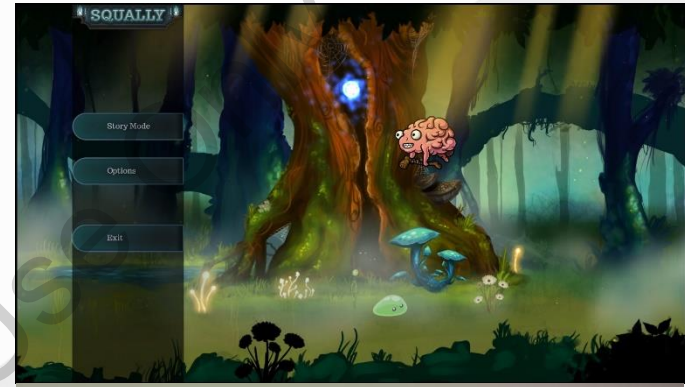
Advantages of C#

Cross Language Support: The code written in any other .NET language can be easily used and integrated with C# applications.

Common Internet Protocols: .NET offers extensive support for XML, which is the preferred choice for formatting information over the Internet.

Simple Deployment: Deployment of C# applications is made simple by the concept of assemblies.

XML Documentation: Comments can be placed in XML format and can then be used as required to document the code.

Introduction to C# 9.0

C# 9.0 offers several new features, some of which include:

Records

- A record type provides immutability supported by 'with' expression.

Relational pattern matching

- It is added to extend pattern matching capabilities to relational operators for comparative evaluations and expressions

Top level statement

- It is used to make programs simpler and by using this feature, main method can be omitted and length of the program can be reduced.

Visual Studio 2019

Visual Studio 2019:

▸ Is a complete set of development tools for building high performance desktop applications, XML Web Services, mobile applications, and ASP Web applications.

▸ Is used to simplify team-based design, development, and deployment of enterprise solutions.

Visual Studio 2019 Environment

- Is an IDE used to ease the development process of .NET applications.
- Has the advantage for all the .NET compatible languages, the same IDE, debugger, Solution Explorer, Properties tab, Toolbox, standard menus, and toolbars are used.
- Provides a single environment to develop .NET applications.
- Supports several programming languages.
- Can be customized.
- Provides built-in browser.
- Executes program with and without a debugger.
- Publishes applications over the internet or onto a disk.
- Provides Dynamic Help feature.
- Provides error notification.
- Provides standard code editor.
- Has a set of visual designer, such as Windows Form Designer, Web Designer, WPF Designer, Class Designer, and Data Designer.
- Has integrated compiler to compile and execute the application.

Visual Studio 2019 Editions

Different editions of Visual Studio 2019 are as follows:

Editions	Description
Visual Studio Enterprise 2019	It provides high productivity and smooth coordination across teams of any size.
Visual Studio Professional 2019	It improves productivity with professional developer tools and services to build applications for any platform (such as Windows, Linux, and so on).
Visual Studio Team Explorer 2019	It provides a free solution for non-developers to interact with Azure DevOps Server and Azure DevOps.
Visual Studio Community 2019	It is a free, fully featured, and extensible solution for individual developers to create applications for Android, iOS, Windows, and the Web.

Elements of Visual Studio 2019 IDE 1-3

Key elements seen in Visual Studio 2019 IDE are as follows:
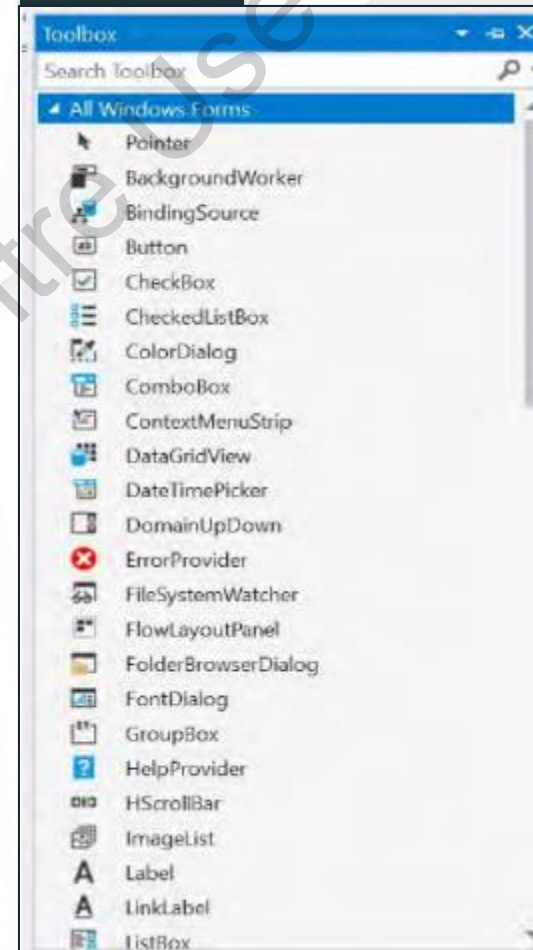
Solution Explorer
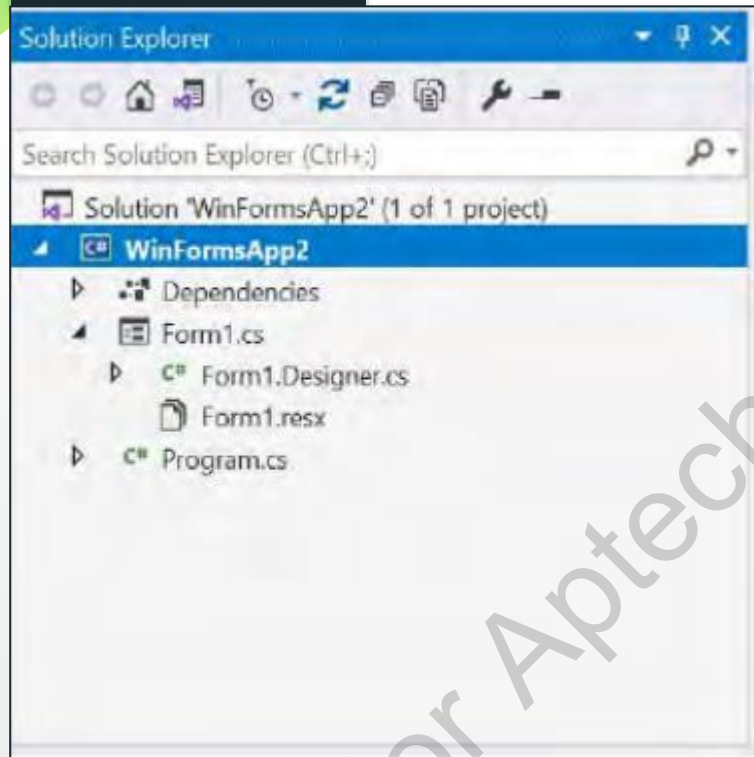
Code Editor
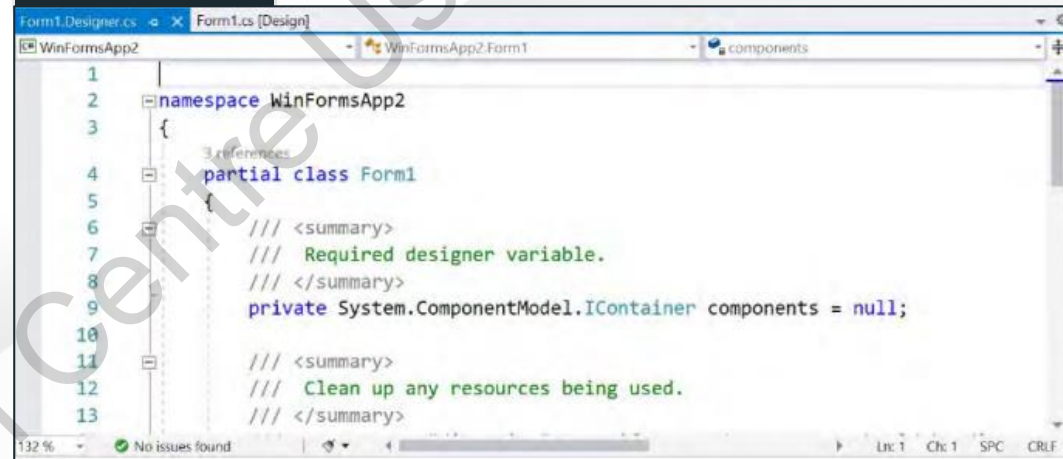
Elements of Visual Studio 2019 IDE 2-3
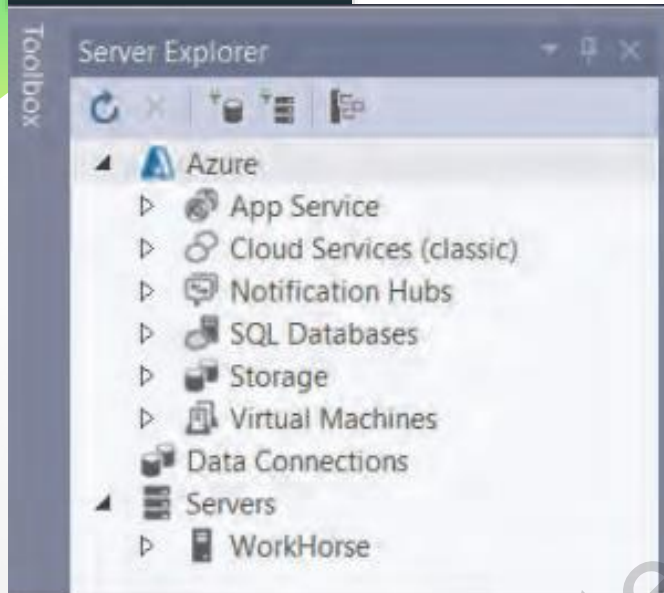

Properties window
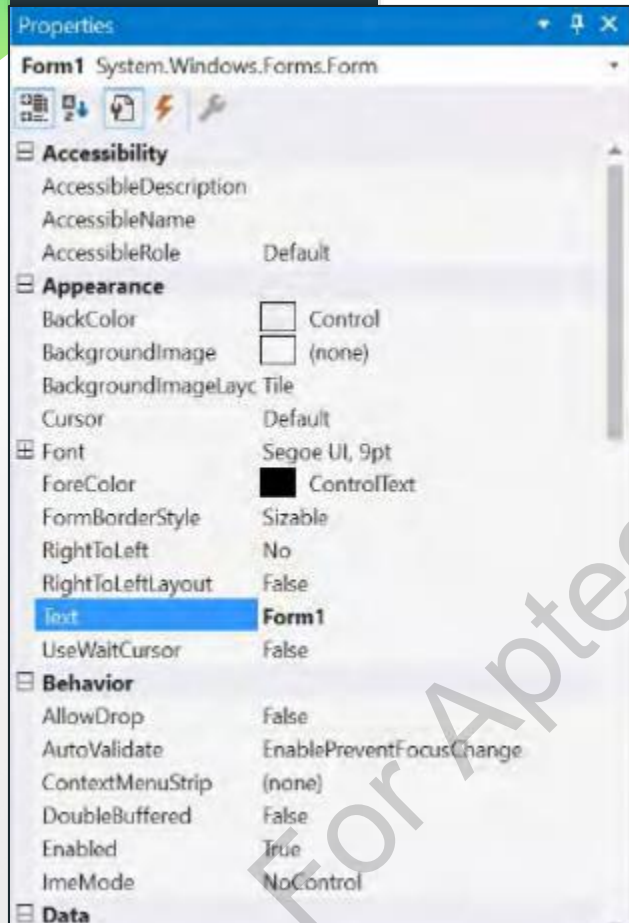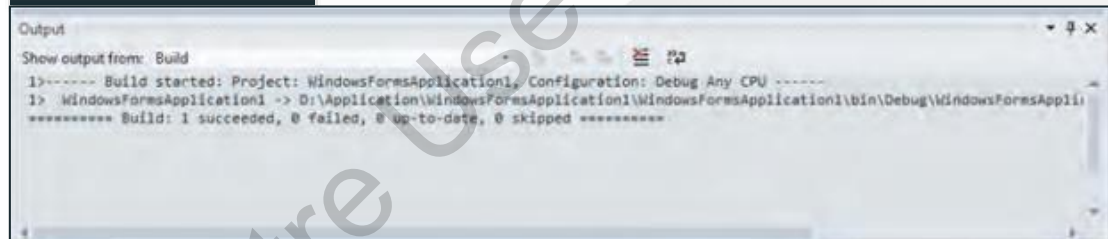

Toolbox

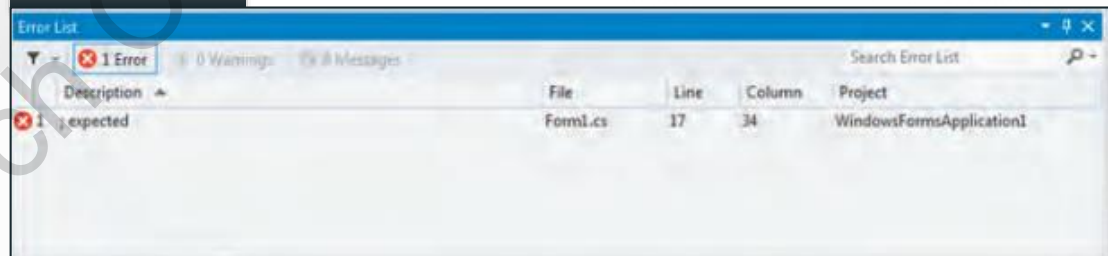# Elements of Visual Studio 2019 IDE 3-3

Server Explorer

Output window:

Error List

Creating an Application in Visual Studio 2019 1-2

To create a new C# console-based project in Visual Studio 2019:

▶ Start Visual Studio 2019.

▶ Select **New** - **Project** from the **File** menu.

▶ Select **C#**, **Windows**, and **Console** respectively in the drop-downs and select **Console Application** in the **New Project** dialog box.

▶ Specify the name and location for the project and click **NEXT**.

▶ Select target framework as .NET 5.0 and click **Create**. Visual Studio 2019 opens the Code Editor with the skeleton code of a class, as shown in following code:

```
using System;
namespace ConsoleApp1
{
  class Program
  {
    static void Main(string[] args)
    {
      Console.WriteLine("Hello World!");
    }
  }
}
```
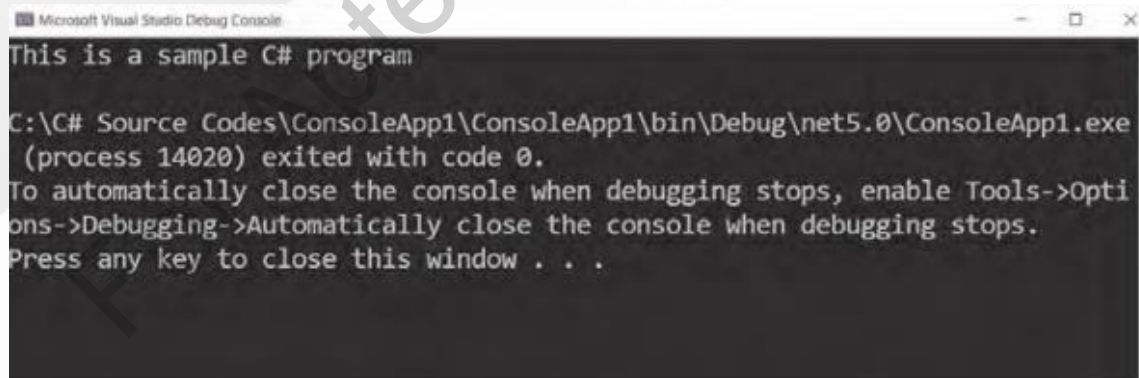
Creating an Application in Visual Studio 2019 2-2

▶ Delete the existing statement after the opening curly brace of the Main(string[] args) method definition and add following statement:

```
Console.WriteLine("This is a sample C# program");
```

▶ Save the project.

▶ Select **Build <application name>** from the **Build** menu. This action will create an executable file (.exe).

▶ From the **Debug** menu, select **Start Without Debugging**.

▶ The project is successfully compiled and notification is displayed. The output of the code is shown in Figure:

Summary

- .NET Framework is a revolutionary platform created by Microsoft for application developers, which includes several tools and technologies that simplify software development.

- .NET Framework 5 is included with Visual Studio .NET 2019.

- Some core components of .NET Framework are: CLI, CLR, DLR, FCL, CTS, and CLS.

- Garbage collection is considered one of the most important features of the managed code.

- The garbage collector of CLR provides automatic memory management during Managed Execution.

- C# has features common to most object-oriented languages and in addition, it has language-specific features, such as type-safety checking, generics, and indexers that make it the preferred language to create a wide variety of applications.

- Visual Studio 2019 provides the environment to develop, debug, test, collaborate, and run applications developed using the .NET Framework.