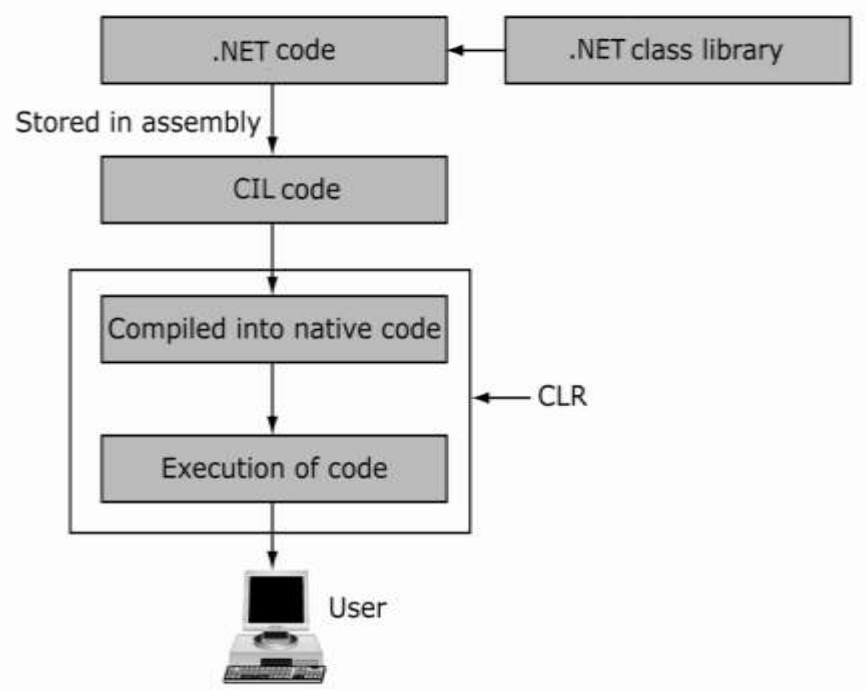
Session: 1

Building Applications Using C#

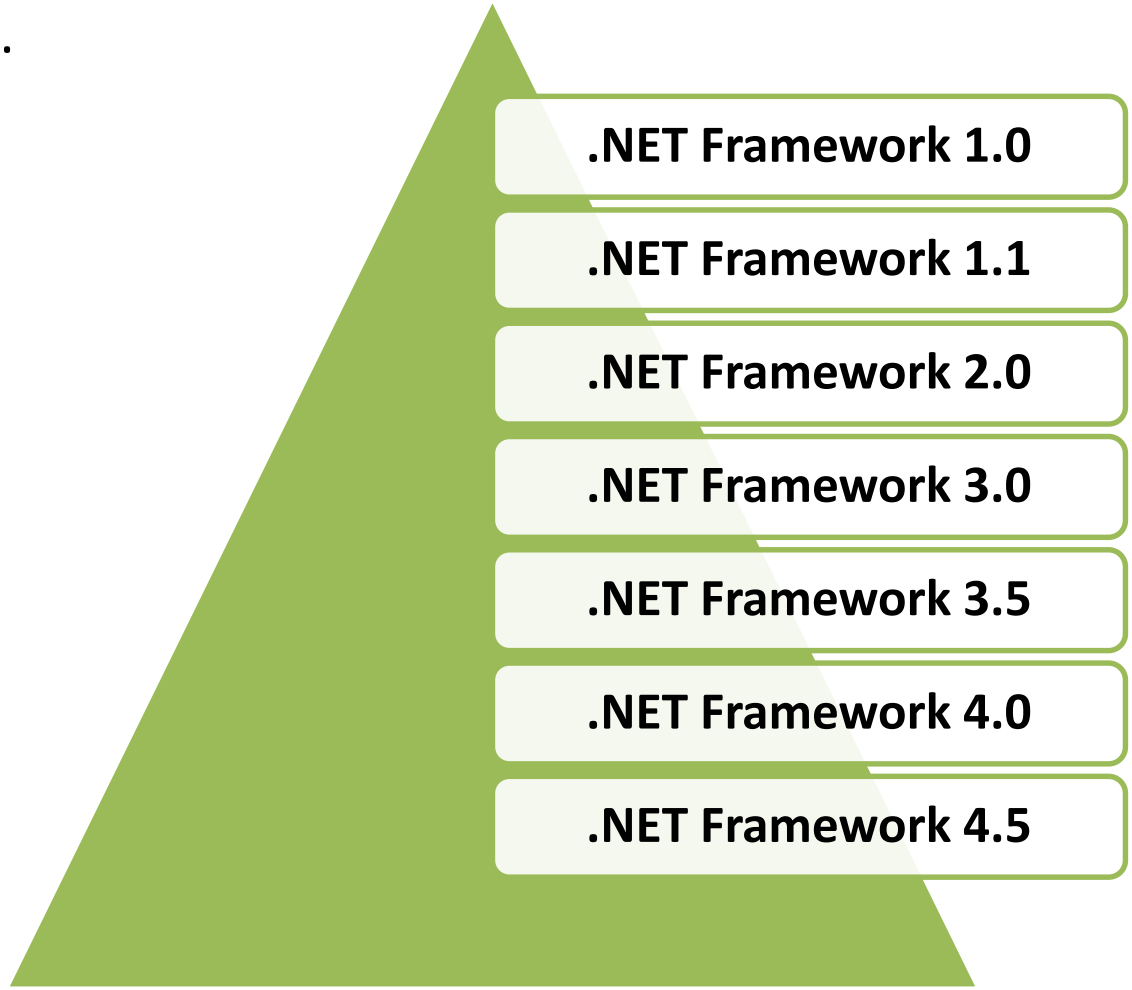
Objectives

- Define and describe the .NET Framework
- Explain the C# language features
- Define and describe the Visual Studio 2012 environment
- Explain the elements of Microsoft Visual Studio 2012 IDE

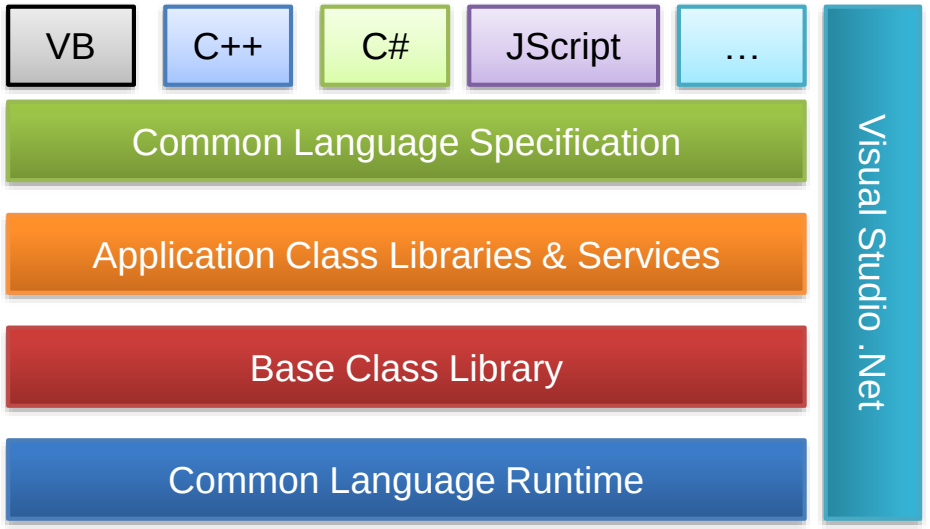
- In traditional Windows applications:
  - Codes were directly compiled into the executable native code of the operating system.
- With .NET framework:
  - Code is compiled into CIL (formerly called MSIL) and stored in a file called assembly.
  - Assembly is then compiled to the native code by CLR



- Microsoft has released different versions of the .NET Framework including additional capabilities and functionalities with every newer version.



- The .NET Framework is an essential component for building and running the next generation applications and XML Web services.
- It is designed to:
  - Provide consistent OOP environment.
  - Provide a code-execution environment that:
    - Minimizes software deployment and versioning conflicts
    - Promotes safe execution of code
  - Provide a consistent development experience across varying types of applications such as Windows apps and Web-apps.



- The two core components of the .NET Framework are:

Common Language Runtime (CLR)

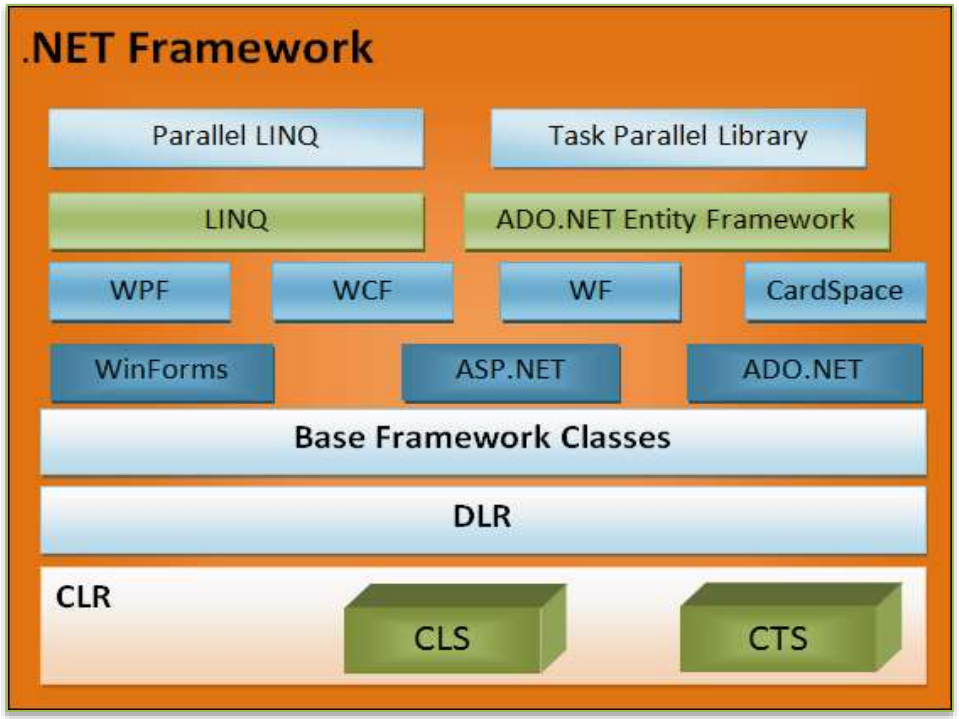
- Is a backbone of .NET Framework
- Performs various functions such as:
  - Memory management, Code execution
  - Error handling, Code safety verification
  - Garbage collection

.NET Framework Class Library (FCL)

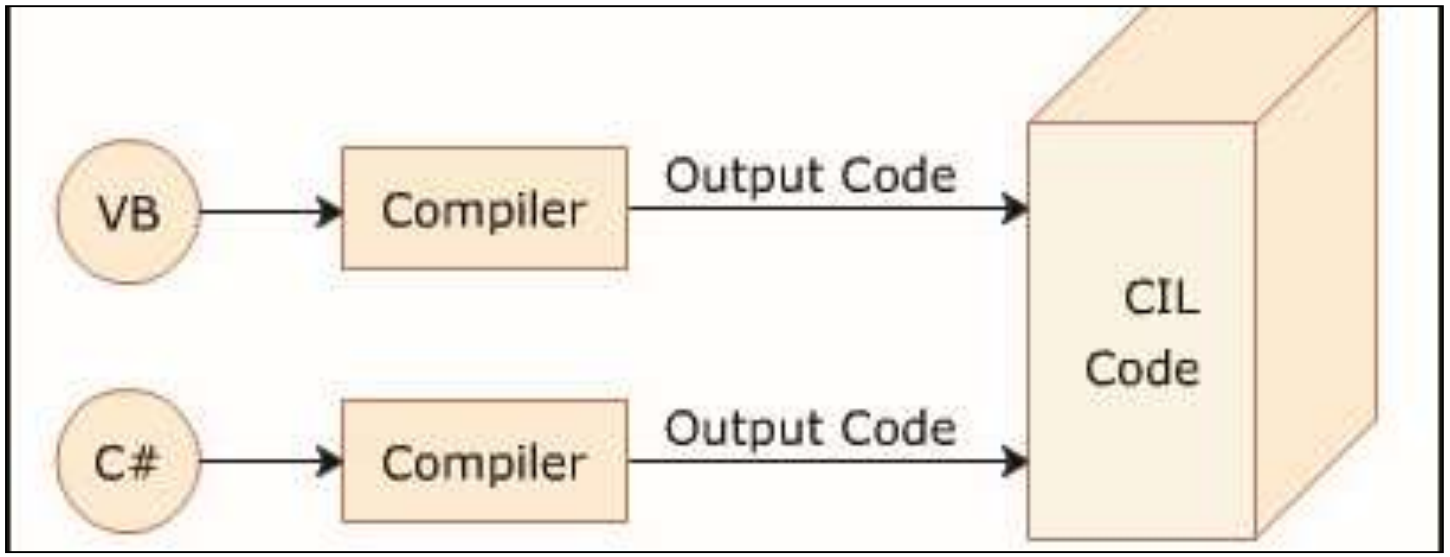
- Is a comprehensive object-oriented collection of reusable types.
- Used to develop applications ranging from traditional command-line to GUI applications that can be used on the Web.

Other Components of .NET Framework

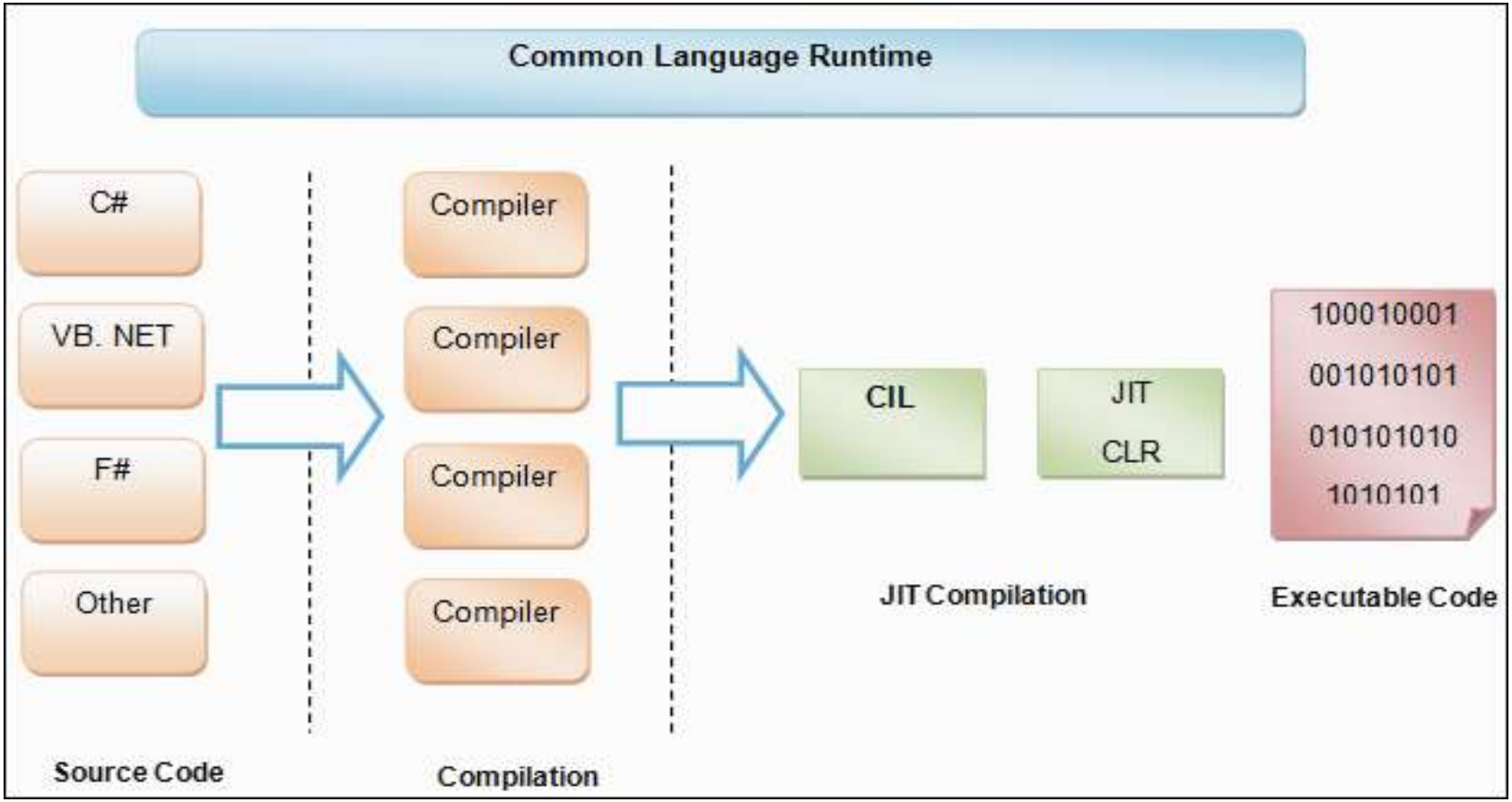
- Following are some other important components:
  - Common Language Specification (CLS)
  - Common Type System (CTS)
  - Base Framework Classes
  - ASP.NET
  - ADO.NET
  - WPF
  - WCF
  - LINQ
  - ADO.NET Entity Framework
  - Parallel LINQ
  - Task Parallel Library



- The following figure depicts the concept of CIL:



- The following figure shows a more detailed look at the working of the CLR:



Visual Studio 2012 Editions

VS Professional 2012

- This is the entry-level edition that provides support for developing and debugging applications, such as Web, desktop, cloud-based, and mobile applications.

Professional with MSDN

- provides all the features of the VS Professional 2012 along with an MSDN subscription.
- includes Team Foundation Server and provides access to cloud, Windows Store, and Windows Phone Marketplace.

Test Professional with MSDN

- targets testers and Quality Assurance (QA) professionals by providing project management tools, testing tools, and virtual environment to perform application testing.

Premium with MSDN

- provides all the features of the combined VS Professional 2012 and VS Test Professional 2012 with MSDN editions.
- supports peer code review, UI validation through automated tests, and code coverage analysis to determine the amount of code being tested.

Ultimate with MSDN

- has all the features of the other editions
- supports designing architectural layer diagrams, performing Web performance and load testing, and analyzing diagnostic data collected from runtime systems.

Languages in Visual Studio 2012

- Visual Studio 2012 supports multiple programming languages such as:
  - Visual Basic .NET
  - Visual C++
  - Visual C#
  - Visual J#
- The **classes** and **libraries** used in the VS 2012 IDE are common for all the languages.
- It makes Visual Studio 2012 more flexible.

- Console applications that are created in C# - run in a console window. This window provides simple text-based output.
- The **csc** (C Sharp Compiler) command can be used to compile a C# program.
- Following are the steps to compile and execute a program:
  1. **Create a New Project**.
  2. **Compile a C# Program**.
  3. **Execute the Program**.

- **Compile a C# Program:**
  - A C# program can be compiled using the following syntax:

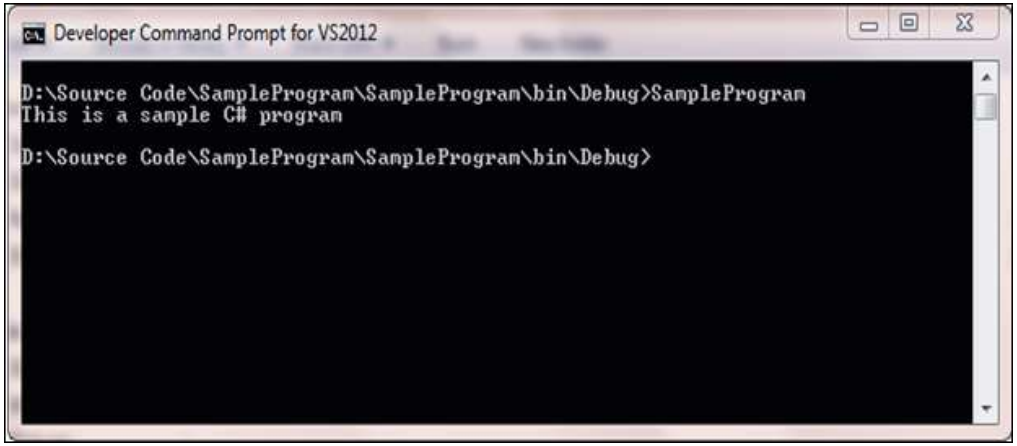
```
csc <file.cs>
```

Example

```
csc SampleProgram.cs
```

  - In the example:
    - **SampleProgram**: Specifies the name of the program to be compiled.
    - This command generates an executable file **SampleProgram.exe**.

- **Execute the Program:**
  - Open the Developer Command Prompt for VS2012, and browse to the directory that contains the `.exe` file.
  - Type the file name at the command prompt.
  - The following figure shows the developer command prompt for VS2012 window:



  - The .exe file is known as **portable EXE** as it contains machine-independent instructions.
  - The portable EXE works on any operating system that supports the .NET platform.

Build and Execute

- The IDE also provides the necessary support to compile and execute C# programs.
- Following are the steps to compile and execute C# programs:
  - **Compiling the C# Program**
    - Select **Build** **<application name>** from the **Build** menu. This action will create an executable file **(.exe).**
  - **Executing the Program:**
    - Select **Start Without Debugging** from the **Debug** menu.

Summary

- The .NET Framework is an infrastructure that enables building, deploying, and running different types of applications and services using .NET technologies.
- The two core components of the .NET Framework which are integral to any application or service development are the CLR and the .NET Framework class library.
- The CLR is a virtual machine component of .NET that is used to convert the CIL code to the machine language code.
- Visual Studio 2012 provides the environment to create, deploy, and run applications developed using the .NET framework.
- Some of the languages supported by Visual Studio 2012 include Visual Basic .NET, Visual C++, Visual C#, Visual J#, and Visual F#.
- C# is an object-oriented language derived from C and C++.