Lecture No 4

Muhammad Siddique

Introduction to .Net Framework

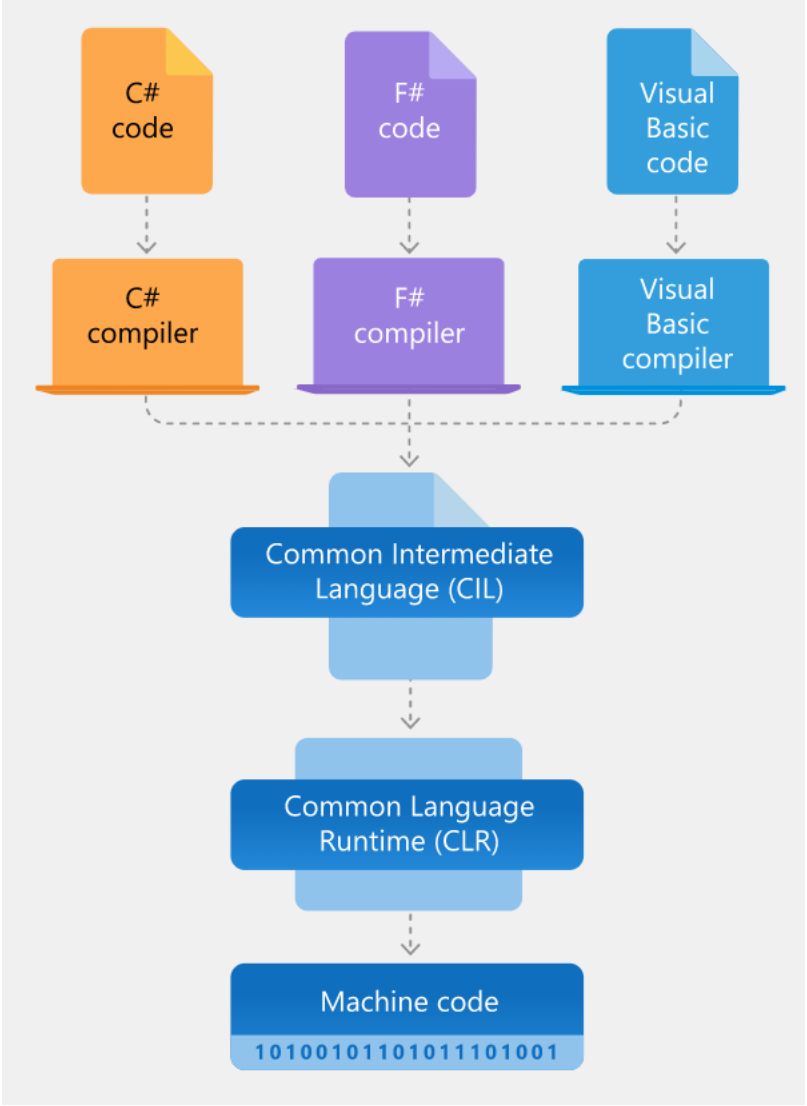
Outline

- Introduction to .Net Frame work.
- Architecture of .Net Framework.
- Benefits of .Net Framework.
- Compiler
- Interpreter
- Hybrid
- Java vs .Net

Introduction to .Net Framework

- .NET Framework is a software development framework for building and running applications on Windows.
- .NET is a developer platform made up of tools, programming languages, and libraries for building many different types of applications.
- .NET Standard is a formal specification of the APIs that are common across .NET implementations. This allows the same code and libraries to run on different implementations.

Architecture of .Net Framework (Diagram)



Architecture of .Net Framework

- .NET Framework is a software development framework for building and running applications on Windows.

- The two major components of .NET Framework are the Common Language Runtime and the .NET Framework Class Library.

  i. The Common Language Runtime (CLR) is the execution engine that handles running applications. It provides services like thread management, garbage collection, type-safety, exception handling, and more.

  ii. The Class Library provides a set of APIs and types for common functionality. It provides types for strings, dates, numbers, etc. The Class Library includes APIs for reading and writing files, connecting to databases, drawing, 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.

Benefits of .Net Framework

- **BENEFITS OF .NET FRAMEWORK ARE as FOLLOWS:**

i. **Memory Management:** In many programming languages, programmers are responsible for allocating and releasing memory and for handling object lifetimes. In .NET Framework applications, the CLR provides these services on behalf of the application.

ii. **Common Type System:** In traditional programming languages, basic types are defined by the compiler, which complicates cross-language interoperability. In the .NET Framework, basic types are defined by the .NET Framework type system and are common to all languages that target the .NET Framework.

Benefits of .Net Framework

iii. **Extensive Class Library:** Instead of having to write vast amounts of code to handle common low-level programming operations, programmers can use a readily accessible library of types and their members from the .NET Framework Class Library. According to estimate, .Net Framework 4.6 consist of 70,000 Classes in Libraries.

iv. **Development Frameworks and Technologies:** The .NET Framework includes libraries for specific areas of application development, such as ASP.NET for web applications, and Windows Communication Foundation for service-oriented applications.

v. **Language Interoperability:** Language compilers that target the .NET Framework emit an intermediate code named Common Intermediate Language (CIL), which, in turn, is compiled at run time by the common language runtime. With this feature, routines written in one language are accessible to other languages, and programmers can focus on creating applications in their preferred language or languages.

Benefits of .Net Framework

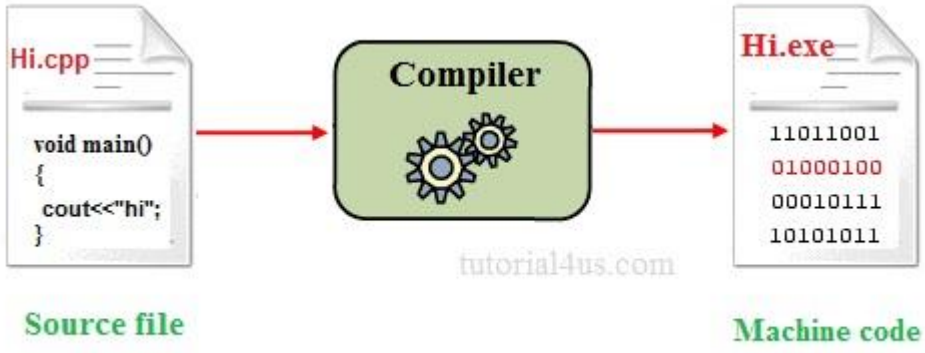
vi. **Version Compatibility:** With rare exceptions, applications that are developed by using a particular version of the .NET Framework can run without modification on a later version.

vii. **Side-by-side execution:** The .NET Framework helps resolve version conflicts by allowing multiple versions of the common language runtime to exist on the same computer. This means that multiple versions of applications can also coexist, and that an application can run on the version of the .NET Framework with which it was built.

viii. **Multi-targeting:** By targeting the .NET Framework Portable Class Library, developers can create assemblies that work on multiple .NET Framework platforms, such as Windows 7, Windows 8, Windows 8.1, Windows 10, Windows Phone, and Xbox 360.

Compiler

- A compiler is a program that reads in as input a program (in some high-level programming language) and outputs machine language code (for some machine architecture).
- The machine language code can subsequently be executed any number of times using different input data each time.
- Example of compiler in C#:

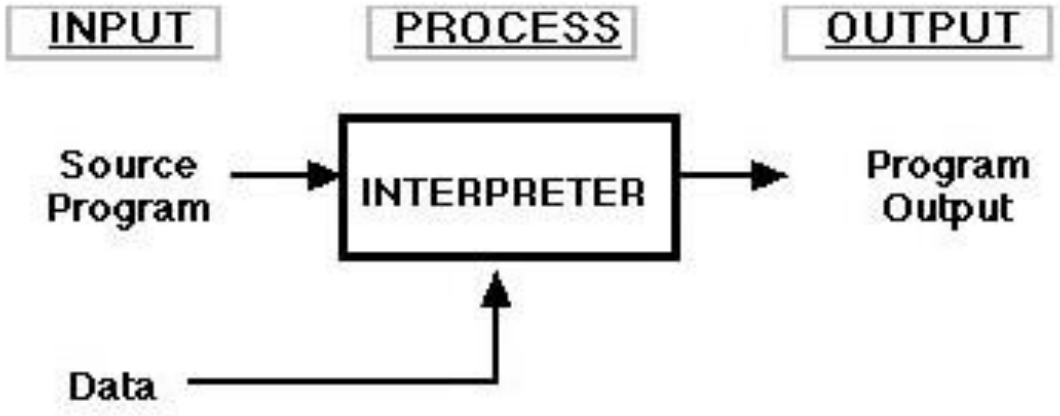


Compiler

- A programmer writes the text of the program using a software program called an editor. The text of a program is referred to as source code and the file is called source file.
- As a example, the Java compiler javac transforms a .java source file into a .class file that is written in Java bytecode, which is the machine language for an imaginary machine known as the Java Virtual Machine.
- C# is compiled is a compiled language as it is compiled into intermediate language (IL), by the c# compiler. This IL is then compiled just-in-time (JIT) as it's needed, into the native assembly language of the host machine.
- Some of the most popular high level programming languages are compiled languages, like C++, C#, Objective – C, Rust and Swift.

Interpreter

- An interpreter is a program that reads in as input a source program, along with data for the program, and translates the source program instruction by instruction.

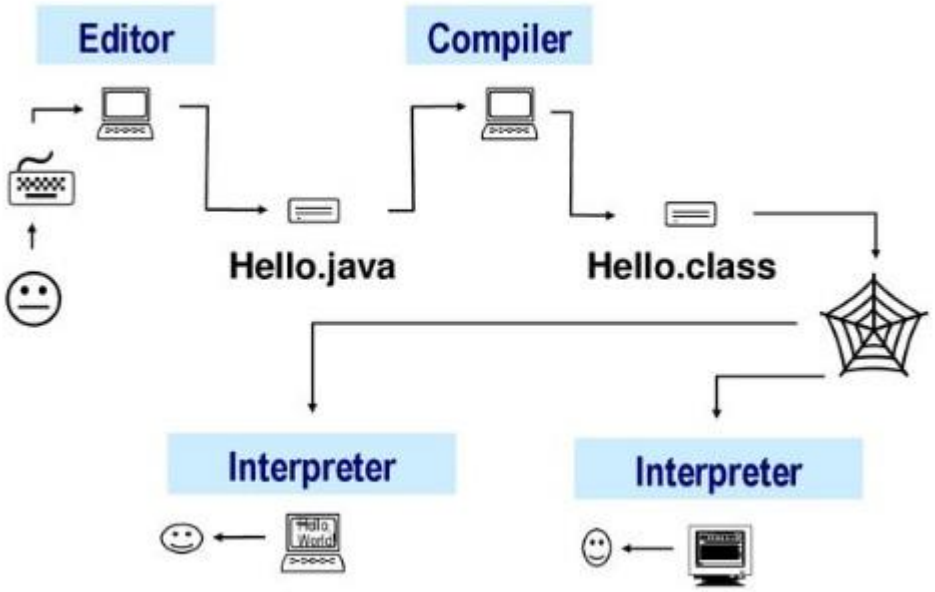


Interpreter

- For example, the Java interpreter java translate a .class file into code that can be executed natively on the underlying machine.
- As a second example, the program Virtual PC interprets programs written for the Intel Pentium architecture (IBM-PC clone) for the PowerPC architecture (Macintosh).
- This enable Macintosh users to run Windows programs on their computer.
- Some of the most popular interpreted languages are JavaScript, Perl and PHP.

Hybrid

- Hybrid approach is a blend of both compiler and interpreter.
- Example: Java uses both compiler and interpreter. The diagram below illustrates Java's compiler + interpreter approach.



Hybrid

- After a programmer uses an editor to enter a simple Java program, names and saves it as Hello, the program is saved as a source file with the file extension .java.
- Then the Java compiler compiles the source file Hello.java, transforms the source file into Hello.class that is written in Java bytecode, which is the machine language for an imaginary machine known as the Java Virtual Machine (JVM).
- Finally, the Java interpreter translate Hello.class into code that can be executed natively on the underlying machine.

Java vs .Net

Parameters	Java	C#
Creation	Designed by Sun Microsystems.	Designed as part of Microsoft's .NET initiative.
Ecosystem	Has a huge opensource ecosystem.	Used to develop software for Microsoft platforms.
Support for generics	It is implemented using erasures and casts added upon compilation into bytecode.	Integrated into the CLI and allows type information to be available at runtime
Support for delegates	Requires use of an interface to achieve similar functionality.	Has delegates which serve as methods that can be called without knowledge of target object.
Checked exceptions	Only has one type of exception	Distinguishes between checked and unchecked exceptions

Java vs .Net

Polymorphism	Invokes the “virtual” keyword in a base class and “override” keyword in a derived class.	Enables polymorphism by default.
Designed for	Java programming language is intended to be run on a Java platform, by the help of Java Runtime Environment (JRE).	The C# programming language is designed to be run on the Common Language Runtime (CLR).
Safety type	Java type safety is safe.	C# type of safety is unsafe.
Built-in Datatype	Built-in data types that are passed by value are called simple types.	Built-in data types that are passed by value are known as primitive types.
Arrays	Arrays in Java are a direct specialization of Object.	Arrays in C# are a specialization of System.

Java vs .Net

Support for conditional compilation.	Java doesn't provide support for conditional compilation	C# supports conditional compilation feature with the help of preprocessor directives.
Support for Goto statement.	Java doesn't support the goto statement.	C# supports the goto statement.
Structure and unions	Java doesn't support structures and unions.	C# supports structures and unions.
Suited for	For concurrency and complex project.	It is mainly suited for game app development projects.
Installation	Requires JDK to run Java.	.Net framework provides a vast library of codes used by C#
Cross-platform support	Java is highly cross-platform with its byte code.	Compared to Java, C# need to improve on this feature.
IDE	Eclipse, NetBeans, IntelliJ IDEA	Visual Studio, MonoDevelop

Java vs .Net

Operator Overloading	No support for operator overloading	C# provides support for operator overloading for multiple operators.
Number of Public Classes	In Java, there is an only a single public class inside source code, or it will display compilation error.	In C#, there are multiple public classes included in the source code.
Control for API	It is controlled by an open community process.	Microsoft controls C# API.
Runtime Environment	Java supports JVM(Java Virtual Machine).	C# supports CLR(Common Language Runtime).
Platform Dependency	Java is a robust and platform independent language.	Code written in C# is windows specific.
Pointers	Java does not support pointers.	In C# you can use pointer only in an unsafe mode.

Java vs .Net

Floating point	Java supports strctfp keyword that means it results for a floating point will be the same for different platform.	C# does not offer support for strictfp keyword. This means that the result of floating point numbers may not be guaranteed to be the same across all platforms.
Famous companies using	Airbnb, Instagram, Spotify, Netflix, etc.	Stack Exchange, Microsoft, Coderus, Docplanner, etc.
Salary Range	The average salary earned by Java Developer is $102,633 per year in the United States.	The average salary for “C# sharp developer” is approximately $108,145 per year.

JVM vs .Net Framework

- Java is a programming language and can work on any operating system.
- Java also provides a Java virtual machine (JVM), which allows code to run on any operating system. NET is actually a framework in the strictest meaning of this word.
- .NET uses ASP.NET to create different applications. Java is a programming language, while .NET is a framework that can use several languages.
- Java and .Net often go hand in hand in the programming world. Both Java and .Net platforms perform similar functions. While both Java vs .NET is often referred to as frameworks, only.