



Summary

Section 2, Lecture 12

So in this section, you learned the basics of C#.

C# vs .NET

C# is a programming language, while .NET is a framework. It consists of a run-time environment (CLR) and a class library that we use for building applications.

CLR

When you compile an application, C# compiler compiles your code to IL (Intermediate Language) code. IL code is platform agnostics, which makes it possible to take a C# program on a different computer with different hardware architecture and operating system and run it. For this to happen, we need CLR. When you run a C# application, CLR compiles the IL code into the native machine code for the computer on which it is running. This process is called Just-in-time Compilation (JIT).

Architecture of .NET Applications

In terms of architecture, an application written with C# consists of building blocks called classes. A class is a container for data (attributes) and methods (functions). Attributes represent the state of the application. Methods include code. They have logic. That's where we implement our algorithms and write code.

A namespace is a container for related classes. So as your application grows in size, you may want to group the related classes into various namespaces for better maintainability.

As the number of classes and namespaces even grow further, you may want to physically separate related namespaces into separate assemblies. An assembly is a file (DLL or EXE) that contains one or more namespaces and classes. An EXE file represents a program that can be executed. A DLL is a file that includes code that can be re-used across different programs.

In the next section, you'll learn about basics of the C# language, including variables, constants, type conversion and operators.