[0140] The CLR and CIL

The Common Language Runtime (CLR)

The CLR is very similar to the "Java Virtual Machine (JVM)" that you used to run your Java programs. Remember that a "Virtual Machine (VM)" is an imaginary computer created by software running inside of a real computer. The key advantage to using a VM to run C# (and Java) programs is that the programs can then run on a wide variety of computer hardware - basic any hardware the can run the VM emulator program. In order words, any platform for which someone (i.e. Microsoft) has implemented the .NET Framework.

Just like an Intel computer runs Intel machine language, the CLR "runs" its own machine language called "CIL" - Common Intermediate Language (which is similar to Java's Bytecode). Because it is running inside of a software-based computer running inside of a virtual machine, .NET programs are "portable" meaning that they can run on any computer that can run .NET.

- Windows, Mac, and Linux now all have .NET and can run .NET programs.
- Potentially phones and other non-Intel devices can too

Here's an example of what IL looks like:

```
// (no C# code)
IL_0000: nop
// int num = 8;
IL_0001: ldc.i4.8
IL_0002: stloc.0
// string str = "Hello";
IL_0003: ldstr "Hello"
IL_0008: stloc.1
// string text = str + num.ToString();
IL_0009: ldloc.1
IL_000a: ldloca.s 0
IL_000c: call instance string [mscorlib]System.Int32::ToString()
IL_0011: call string [mscorlib]System.String::Concat(string, string)
IL_0016: stloc.2
```

This code is very similar to the machine language instructions that actual CPUs (like Intel microprocessors) execute directly however, unlike actual machine language, when CIL is "executed" it is actually ***interpreted*** by the .NET CLR.

The Just-In-Time (JIT) Compiler

The Just-In-Time (JIT) compiler (aka "the Jitter") is a performance optimization feature that automatically speeds up programs by converting heavily used CIL code into native machine code (i.e., Intel machine code) each time the program is run.