**C# .NET Notes**

Introduction

* C# is a statically typed, generally-purpose, OOP, programming language
* Microsoft created C# and C# is the main language used with the .NET Framework.
* Before C#, we had C and C++. When we compile C or C++, the code is translated to machine code for the machine it’s running on. However, different machines have different hardware and operating systems. Thus, C or C++ code on one machine might not run on another machine. Thus, Microsoft wanted to solve this issue which is why they created C#. they made it so that C# code is not compiled into machine code. Instead, it is translated into an intermediate language called IL code (Intermediate Language Code). This IL code is independent of the hardware it’s running on. We then translate the IL code into machine code using the CLR via the .NET framework (more on this later).

.NET

* .NET is an ecosystem and an open-source developer platform for building different types of apps.
* A developer platform is the combination of languages and libraries that you use.
* You can use many languages with .NET such as C#, F#, Visual Basic, etc.
* Platforms are ways we can run .NET. Examples of platforms are:
  + .NET Core: (runs anywhere) Windows, Linux, macOS
  + .NET Framework: websites, services, and apps on Windows
  + Xamarin/Mono: a .NET for mobile
* .NET Core, .NET Framework, Xamarin/Mono are multiple examples of .NET platforms. This can be confusing so there is something called the .NET standard which is a shared set of libraries for the above. So, we only have to think about the one .NET standard even though we might be developing across many .NET platforms.

.NET Framework

* A framework for building applications on Windows.
* This framework consists of two components the CLR (Common Language Runtime) and the Class Library (sometimes called the FCL for framework class library).
* CLR
  + The CLR is the execution engine that handles running applications.
  + It translates IL code into machine code. This process of translating IL code into machine code is called just-in-time complication or JIT. With this architecture, you can write a C# application and we know that our application will successfully run across different machines if they have the CLR.
  + All .NET programs are executed by the CLR, including C# programs.
  + Diagram

    Description automatically generated
  + It provides services like thread management, garbage collection, type-safety, exception handling, and more.
* 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.