.NET- platform for many languages

C# is most popular

Other languages:

F#, VB.NET, Python, Java, R, Typscript/JS

.NET CLI- common language infrastructure OR command line interface

.NET code (.cs file) 🡪 c# compiler(dotnet build, msbuild, csc(c# compiler)) 🡪 (common) intermediate language (IL- intermediate language, CIL- common intermediate language, MSIL- Microsoft intermediate language) turns into a .dll or .exe file for running. These are in bytecode language, projects are called assemblies, which are the instructions for the VES(virtual execution system), also referred to as CLR(Common Language Runtime) 🡪 VES, operates as a JIT compiler (just in time compiler), converts code into native instructions for the CPU as they are needed instead of all at once.

Common Type System (CTS)- allows primitives from different languages to execute the same on the VES

Base Class Library (BCL)- is lists and data structures built into the language

Build is .NET code and CIL

Run is VES and CPU

.NET Framework- the original .NET platform, windows only

.NET Core- current platform, truly cross platform

Mono- a port of .NET Framework for mac and linux

.NET provides us with interoperability, cross-platform/platform independent, architecture-independent (32 and 64 bit runs the same), exception handling, typing, managed environment(garbage collection)

.NET standard is not its own implementation but is the commonality between both .NET Core and .NET Framework

4 major principles of OOP

Abstraction- separation between needed functionality and implementation details, the functionality is accessible without needing to worry about the implementation. Ex. Properties, methods, Common types

Polymorphism- the possibility of many implementations behind a common contract. Ex. Method overriding, method overloading

Encapsulation- Restricting access to information within objects. Ex. Objects, access modifiers.

Inheritance- the ability for a class to take and extend behavior from another class.