Basics

18 June 2022 09:28

.NET: It is a software *platform* for building and executing runtime *managed* applications on different operating systems (Windows, Linux, macOS, Android and iOS). The programing environment of .NET consists of

1. **Common Language Runtime** (CLR) - It specifies how data is structured and code is compiled for a .NET application and it handles execution of such compiled (managed) code on top of the services provided by its host. It implements
   (a) **Common Type System** - A .NET data type is either a *value type* whose data can be directly accessed from its identifier or a *reference type* whose data can only be accessed through an indirection from its identifier. The CLR supports (fourteen) *primitive* value types and offers a *unified object-oriented* model for implementing *user-defined* reference and value types.
   (b) **Virtual Execution System** - A group of related .NET data types are compiled into a single unit of deployment called an *assembly* (DLL file) which contains the *meta-data* (machine readable description) of those types along with *intermediate language opcodes* (machine neutral instructions) of their implemented methods. The CLR loads the assemblies required by the executing application and translates the IL opcodes of a method into its equivalent machine instructions *just-in-time* of its invocation.

2. **Base Class Library** (BCL) - It is a framework (Microsoft.NETCore.App) of assemblies consisting of types required by a .NET application for consuming the services offered by the following in a portable manner
   (a) **Runtime** which includes support for built-in types, reflection and native interop.
   (b) **Platform** which includes support for multithreading, file i/o and communication sockets.

3. **C# Programming Language** - It is a high-level programming language (pronounced as See Sharp) designed specifically for coding applications and libraries which target the CLR. It has following important features
   (a) It offers a C++ like but more expressive syntax based on the common type system with opt-in support for pointers and fixed-size buffers.
   (b) It is primarily object oriented based on common root single state inheritance model with added support for generic and functional programming.