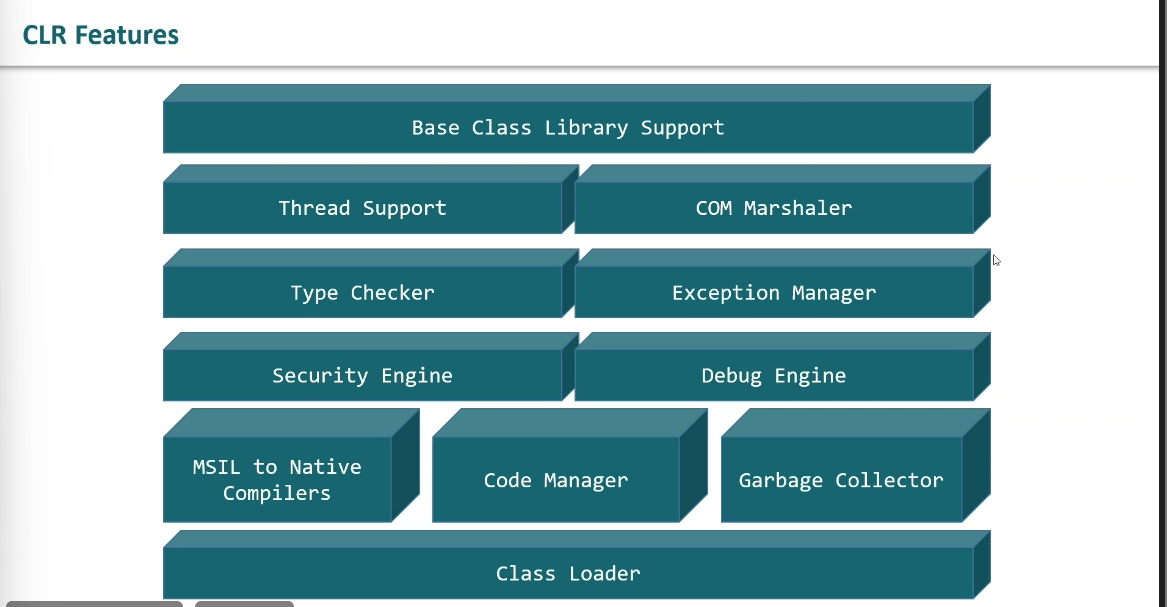
CLR- Common Language Runtime

CLR in .NET Framework is the virtual machine component that enables execution of various languages. This is done by providing various services.

CLR implements the Virtual Execution System (VES)- runtime system to manage code execution environment.

Architecture of CLR



1. Base Class Library Support- It contains multiple libraries that provide various features for .NET programming.

Eg Collections, DataType etc.

1. Thread Support- It is used for managing parallel execution of multiple threads.

System.Threading class is used as parent class for this.

1. COM Marshaller- Component Object Model (COM) , it is used for communication with the COM component in .NET app.

Provides interoperability to COM.

1. Type Checker- Used to provide type safety by using Common Type System (CTS) and Common Language Specification (CLS) that helps in verifying types in app.
2. Exception Manager- Used to handle exception irrespective of the language (present in .NET).

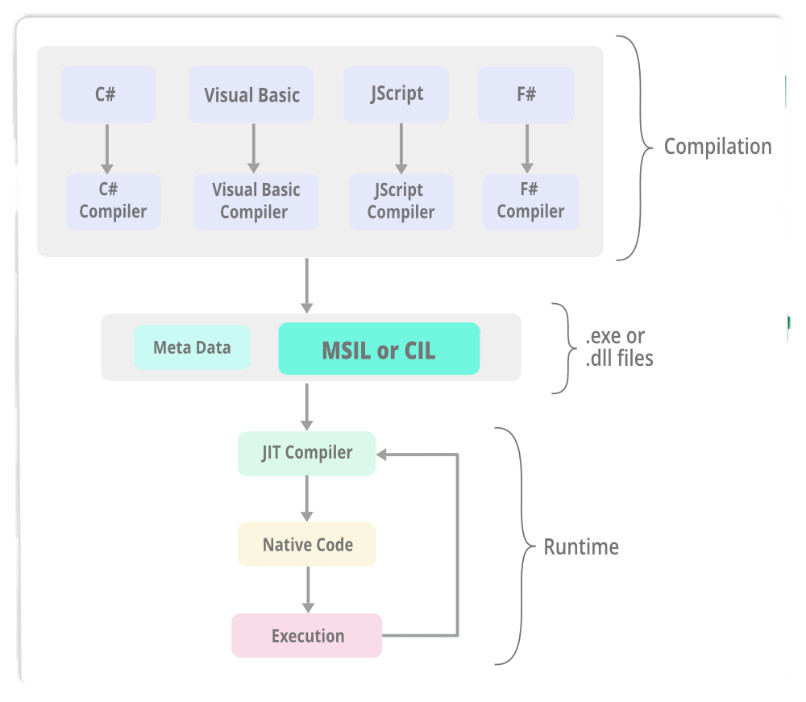
If there is a catch block to handle the exception in case of occurrence else when occurring, it terminates the application.

1. Security Engine- It handles security permissions at different levels like code, folder, machine. To check code safety.
2. Debug Engine- Used to debug during runtime.
3. JIT Compiler- (Just in Time) in CLR converts Microsoft intermediate language into machine code specific to the system environment. JIT is used for managing execution of .NET program irrespective of the program language.

The MSIL file is stored for subsequent calls if required.

1. Code Manager- It manages the .NET code i.e. Managed Code.
2. Garbage Collector- In CLR garbage collector automatically manages the memory. It removes the memory when it’s no longer needed.
3. Class Loader- Used to load class on runtime and manages the meta data.

Execution Process in CLR



* The source code is converted into the MSIL by a language-specific compiler in the compile
* A Common Language Infrastructure (CLI) assembly is created by assembling the MSIL. This assembly is basically a compiled code library that is used for security, deployment, versioning, etc. and it is of two types i.e. process assembly (EXE) and library assembly (DLL).
* The JIT compiler then converts the Microsoft Intermediate Language(MSIL) into the machine code that is specific to the computer environment that the JIT compiler runs on.
* The machine code obtained using the JIT compiler is then executed by the processor of the computer.