INTRODUCTION TO .NET ASSIGNMENT

Q1:- Demonstrate the process of conversion of Source code into the native machine code in .Net framework with the help of a flowchart.

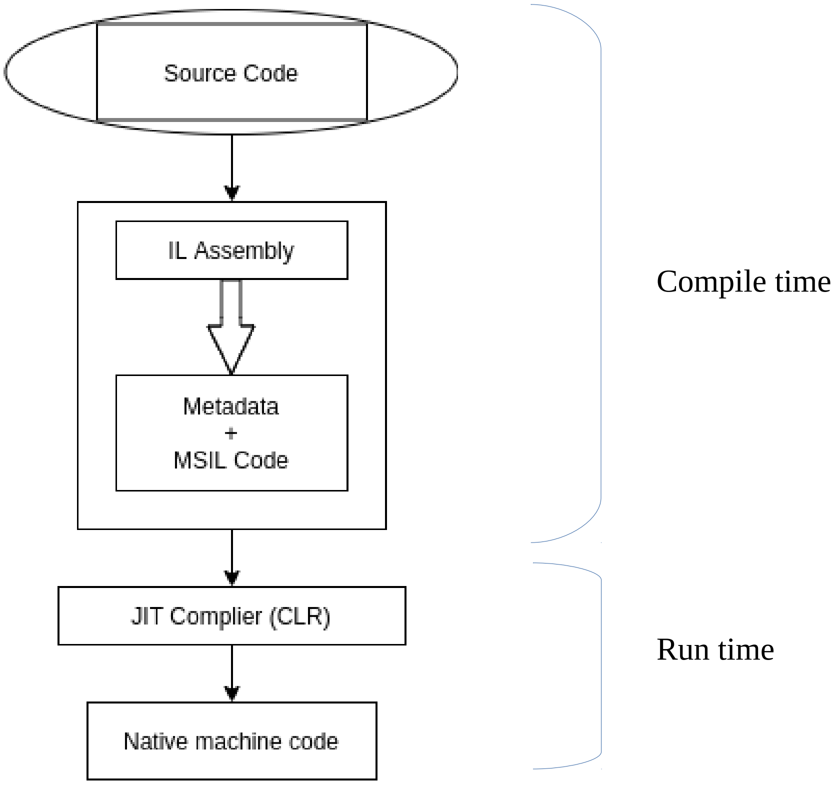
Sol:- Source code is the code that we write to implement business solutions.
The code execution process involves two steps:
1) Compiler time process
2) Runtime process

Compile time process:

- The .net framework has one or more language and third-party compilers. Any f the compilers translate the source code into Microsoft Intermediate Language code.
- In other words, we can say that compiling translates source code into MSIL and generaes the required metadata which is also known as "Intermediate Language code" or "Common Intermediate Language code".

Run time process:

- The Common Language Runtime(CLR) includes a JIT compiler which converts MSIL code to native code which is then executed by the operating system.
- In this process, the JIT compiler of the CLR uses the metadata generated by the .net complier and converts MSIL code into native code.



Conversion of source code to native machine code

Q2:- Explain in detail the CTS and how the .net framework implements CTS.
Sol:- CTS stands for Common Type System.

- CTS defines a collection of data-types which are used and managed by the runtime. It is a standard for defining and using data types in the .net framework.
- CTS provides the types in .net framework with which .net applications, components and controls are builts in different programming languages and ensures that different .net languages can interact with each other.
- All CTS data type derive from a common data type called object. Therefore, they share some common functionality and one type can easilt be converted to another.
- CTS divides the available data types into two categories Value types and reference types.
  - Value types includes Structures, enums, Integer, float, char, bool, doubles, short, long, byte .
  - Reference types includes Classes, Interfaces, Delegates, Arrays, Objects and Strings.
- Suppose we have two variables written in different languages, then CTS will take the values and convert the different variables to a common data type for both programming languages so that bot languages can communicate with each other easily.

Q3:-Name at least 3 runtime services provided by CLR and explain their role in .net framework.
Sol:- CLR provides various services. Some of them are as follows:
1) Garbage Collector: Garbage Collector provides automatic memory management. It handles automatic memory management and it will release memory of unused objects in an application.

2) Code Manager: CLR manages code. When a .net application is complied, we generate a MSIL code. All .net code is IL code known as managed code, because the CLR manages it.

3) Common Language Specification: CLS defines a set of rules and standards which languages use to be compatible with .net languages. CLR provides the CLS service for communication of objects written in different .net languages.

Q4:- What are the differences between Library vs DLL vs .Exe? Explain.
Sol:- A **.lib** is a library of functions that are statically linked to a program i.e., they are not shared by other programs. Each file linked with the .lib file contains the complete code. Libraries are used to for code reusability also.

A **.dll** file is used for Dynamic Link Library. It is a form of file which contains instructions that other files can use to do several specific things i.e., multiple programs can share the skills perogrammed into a document. A dll cannot run itself rather is used as a supportive file for other applications.

A **.exe** file is an executable file and is not a supportive file. It itself is an application and can run individually with the help of an entry point called main. A program containing main creates a .exe file after compilation. For each executing exe file a seperate process is created by the operating system while dll can run on the same process created for an exe.

Q5:- How does CLR in .net ensure security and type safety? Explain.
Sol:- **Type Safety:** A type-safe code only accesses the memory locations it is authorized to access. During the JIT compilation, an optional verification process examines the metadata and MSIL code of a method to verify they are thread safe.

Type safety plays import role in assembly isolation and security. When a code is thread safe, the CLR completely isolate assemblies with each other which ensures that assemblies cannot affect each other. In this way CLR ensures thread safety.

Security: CLR provides us code access security(CAS) . CAS grants rights to program depending on the security configuration of the machine. To ensure securiy the CLR denies several operations not present in the security configuration. CAS tales care that the code runs under the machine security configuration environment.