**1.Demonstrate the process of conversion of Source code into the native machine code in .Net framework with the help of a flowchart.**

Ans.

a)Suppose you have written a C# program and save it in a file which is known as the Source Code.

b)Language specific compiler compiles the source code into the MSIL(Microsoft Intermediate Language) which is also know as the CIL(Common Intermediate Language) or IL(Intermediate Language) along with its metadata. Metadata includes the all the types, actual implementation of each function of the program. MSIL is machine independent code.

c) Now CLR comes into existence. CLR provides the services and runtime environment to the MSIL code. Internally CLR includes the JIT(Just-In-Time) compiler which converts the MSIL code to native machine code which further executed by CPU. CLR also uses the .NET Framework class libraries. Metadata provides information about the programming language, environment, version, and class libraries to the CLR by which CLR handles the MSIL code. As CLR is common so it allows an instance of a class that written in a different language to call a method of the class which written in another language.



**Flowchart**

**2.Explain in detail the CTS and how the .net framework implements CTS**

a)Every programming language has its own data type system, so CTS is responsible for understanding all the data type systems of .NET programming languages and converting them into CLR understandable format which will be a common format.

b) There are 2 Types of CTS that every .NET programming language have :

i)Value Types: Value Types will store the value directly into the memory location. These types work with stack mechanism only. CLR allows memory for these at Compile Time.

ii)Reference Types: Reference Types will contain a memory address of value because the reference types won’t store the variable value directly in memory. These types work with Heap mechanism. CLR allots memory for these at Runtime.

**3.Name at least 3 runtime services provided by CLR and explain their role in .net framework**

CLR provides multiple services to execute processes, like memory management service and security services. CLR performs multiple tasks to manage the execution of .NET applications. Following responsibilities of CLR are given below:

1)Automatic memory management

2)Code access security

3)Garbage collection

4)JIT compilation

1) Automatic memory management

CLR calls various predefined functions of .NET framework to allocate and de-allocate memory of .NET objects. So that, developers need not to write code to explicitly allocate and de-allocate memory.

2) Code access security

CLR allows access to code to perform only those tasks for that it has permission. It also checks user’s permissions using authentication and configuration files of .NET applications.

3) Garbage collection

GC is used to prevent memory leaks or holes. Garbage collector of CLR automatically determines the best time to free the memory, which is allocated to an object for execution.

4) JIT compilation

JIT stands for Just In Time. It is also an important part of Common Language Runtime (CLR), JIT compiler converts MSIL code to targeted machine code for execution.

**4.What are the differences between Library vs DLL vs .Exe? Explain**

A library file is just a collection of related obj files, much like putting obj files in a directory. That is essentially what a lib file is, a library of obj files. For a static link, all of the obj files that an executable uses are combined into one file.

An .exe is a very common file type. The .exe file extension is short for “executable.” These files are most commonly used on Windows computers to install or run software applications..exe is a stand alone executable file, which means it can executed directly.

. dll is a reusable component which cannot be executed directly and it requires other programs to execute it. An exe is an executible program whereas A DLL is a file that can be loaded and executed by programs dynamically.

**5.How does CLR in .net ensure security and type safety? Explain**

CLR analyses th MSIL instruction whether they are safe or unsfe .Thus the CLR increases security.Also,CLR uses the delegates instead of function pointers .So,by this CLR enhances the type safety and security.