**Enterprise Application Development**

**(Home Work 1)**



**Submitted to: Dr. Shuja Ul Rehman**

**Submitted by: Hafiz Muhammad Ali**

**Roll no: BSEF18A002**

**Q1: What is IL and how IL code is converted into machine executable code?**

**Answer:** IL stands for Intermediate Language. After compilation of source codes of C#, F#, or VB, compilers generate this file. Its code could not be executed directly on any operating system. It’s independent of any CPU. If we want to run this code on any operating system we have to generate a native code, and for this purpose we use CoreCLR. IL code is converted into Native Code parts by parts by using RyuJIT.

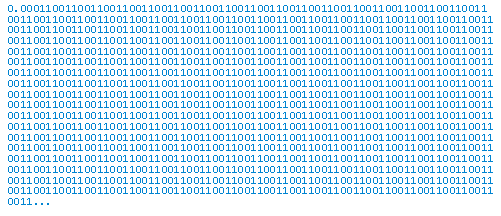
**Q2: Differentiate between .NET Core and .NET Frameworks. (Minimum 4 Points)**

|  |  |
| --- | --- |
| .Net Core | .NET Frameworks |
| * Independent of a specific operating system. Application runs anywhere (windows, linux, macOS) | * Application written in it runs only on windows OS. |
| * It includes a cross-platform implementation of the CLR known as CoreCLR and a streamlined library of classes known as CoreFX. | * It includes a Common Language Runtime (**CLR**) which manages the execution of code, and a Base Class Library (**BCL**) or (**FCL)** framework class library, which provides a rich library of classes to build applications from. |
| * It changes frequently because it can be deployed side by side with an app and will not affect other .NET Core apps on the same machine. | * It’s updated infrequently. |
| * .NET Core is both open source and cross-platform. | * .NET Framework is what it is. The current version of .NET Framework, 4.8, is supposed to be the last version of .NET Framework. There will be no more new versions of .NET Framework planned in the future. |

**Q3: Why Decimal data type variable takes more space and store less data than double type variable? Please explain it with example.**

**Answer:** Double data type has 8 bytes in memory whereas Decimal data type has 16 bytes in memory but Decimal data type variable takes more space and store less data than double type variable because Decimal data type’s value is stored in memory in Decimal formats whereas Double data type’s value is stored in memory in binary notation (0 or 1). If you want to compare values than don’t use double data type as in binary notation a number can be store in different representations.

The answer is that most decimals have infinite representations in binary. Take 0.1 for example. It’s one of the simplest decimals you can think of, and yet it looks so complicated in binary:



Decimal 0.1 In Binary (To 1369 Places)

The bits go on forever; no matter how many of those bits you store in a computer, you will never end up with the binary equivalent of decimal 0.1.

But when we store 12.34 in decimal data type it stores like 1234 with decimal shift 2 points left. Thus in decimal data type we have to store decimal points position. Therefore Decimal stores less data in it than double data type.

**Q4: The default values of**

1. **Int** >> 0

2. **uint** >> 0

3. **Double** >> 0.0D

4. **Float** >> 0.0F

5. **Decimal** >> 0.0M

6. **Object** >> NULL

**Q5: Please find out the errors if they have in these snippets and rewrite the code.**

1: static void Main(string[] args)

{

string String = null;

int? numbers = 85;

String = $"Ahmad got [numbers] marks/n";

object obj = String;

obj = obj.ToLower();

Console.writeLine(obj);

}

**Correct code:**

static void Main(string[] args)

{

string String = null;

int numbers = 85;

String = $"Ahmad got {numbers} marks\n";

object obj = String;

obj = ((string)obj).ToLower();

Console.WriteLine(obj);

}

**2: static void Main(string[] args)**

{

int age = Console.Read();

Console.writeLine(age);

}

**Correct code:**

static void Main(string[] args)

{

string age = Console.ReadLine();

Console.WriteLine(age);

}

**Q6: Differentiate between var and dynamic with an example.**

**Answer:**

**Var:** Var is checked on compile time. Therefore its type is decided on Compile Time. If you know the type of data assigned to it then use Var data type. Its type is decided on compile time so the object of var data type must be initialized at the time of its declaration.

*static void Main(string[] args)*

*{*

*var age = 22;*

*age = “Ali Uppal”;*

*/\* compiler gives an error on compile time as its type is already decided as an integer, so we can’t change its type here.. \*/*

}

**Dynamic Data Type:** Dynamic data type is checked on Run Time. Therefore its type is decided on Run Time. If you don’t know the type of data assigned to it then use Dynamic data type. Its type is decided on run time so there is no need to initialize it on its declaration time but it reduced the efficiency of our program.

*static void Main(string[] args)*

*{*

*dynamic age = 22;*

*age = “Ali Uppal”;*

*// compiler doesn’t give an error on compile time as its type is decided run time*

*}*

**Q7: Write a program that displays the ascii value of a key when pressed by user.**

static void Main(string[] args)

{

Console.Write("Please Enter any key: ");

ConsoleKeyInfo key = Console.ReadKey();

int asciii = key.KeyChar;

Console.WriteLine(

format: "Key: {0}, Ascii: {1} ",

arg0: key.Key,

arg1: asciii

);

}

**Q8: Write a program that displays the table of a number X to the length N. Your program must get these values as arguments in the main function. You need to use formatted output in your program to get full marks.**

static void Main(string[] args)

{

Console.WriteLine(args.Length);

int X = Convert.ToInt32(args[0]);

int N = Convert.ToInt32(args[1]);

Console.WriteLine($"The Table of {X} up to multiples of {N} is given below");

for (int i = 1; i <= N; i++)

{

Console.WriteLine(

format: "{0} x {1} = {2}",

arg0: X,

arg1: i,

arg2: X \* i

);

}

}

**Q9: Write a program that takes 5 Fruits names along with their prices in command line arguments and display them in proper output formatting.**

static void Main(string[] args)

{

Console.WriteLine("The fruits detail are given below");

for (int i = 0; i < args.Length; i++)

{

Console.WriteLine($"{args[i++]} \t {args[i]:C}");

}

}