Question 1: - Explain difference between .NET and C#?

1. C# is a programming language, whereas .NET is the framework on which the language is built.

Question 2:- .NET Framework vs .NET Core vs .NET 5.0?

A)

|  |  |
| --- | --- |
| Platform independent. Works with Windows, macOS, Linux | Works only on Windows environment. |
| You may use Razor pages to create Web UI. | Uses traditional Web forms to create web pages. |
| You can develop ASP.Net Core based applications in Visual Studio, Visual Studio for Mac, Visual Studio Code. | Here, we have Visual Studio only. |
| Uses .NET Core runtime | It uses .NET framework runtime. |

Question 3 :- What is IL ( Intermediate Language) Code ?

1. Intermediate language (IL) is an object-oriented programming language designed to be used by compilers for the .NET Framework before static or dynamic compilation to machine code. The IL is used by the .NET Framework to generate machine-independent code as the output of compilation of the source code written in any .NET programming language.

Question 4 :- What is the use of JIT ( Just in time compiler) ?

1. The JIT compiler translates the MSIL code of an assembly to native code and uses the CPU architecture of the target machine to execute a .NET application.It also stores the resulting native code so that it is accessible for subsequent calls.

Question 5 :- Is it possible to view IL code ?

1. Choose ReSharper | Windows | IL Viewer in the main menu.

Question 6 :- What is the benefit of compiling into IL code ?

1. In addition to SLaks's answer, compiling to IL enables a degree of cross-language interoperability that generally doesn't exist in interpreted languages.

Question 7 :- Does .NET support multiple programming languages ?

1. Yes

Question 8 :- What is CLR ( Common Language Runtime) ?

1. Question 9 :- What is managed and unmanaged code ?  
   Question 10 :- Explain the importance of Garbage collector ?  
   Question 11 :- Can garbage collectors claim unmanaged objects ?  
   Question 12 :- What is the importance of CTS ?  
   Question 13 :- Explain CLS ?  
   Question 14 :- Difference between Stack vs Heap ?  
   Question 15 :- What are Value types & Reference types?  
   Question 16 :- Explain boxing and unboxing ?  
   Question 17 :- What is the consequence of boxing and unboxing ?  
   Question 18 :- Explain casting, implicit casting and explicit casting ?  
   Question 19 :- What can happen during explicit casting ?  
   Question 20 :- Differentiate between Array and ArrayList ?  
   Question 21 :- Whose performance is better array or arraylist ?  
   Question 22 :- What are generic collections ?  
   Question 23 :- What are threads (Multithreading)?  
   Question 24 :- How are threads different from TPL ?  
   Question 25 :- How do we handle exceptions in C#(try/catch)?  
   Question 26 :- What is the need of finally?  
   Question 27 :- Why do we need the out keyword ?  
   Question 28 :- What is the need of Delegates ?  
   Question 29 :- What are events ?  
   Question 30 :- What's the difference between Abstract class and interface ?  
   Question 31  :- What is a delegate and How to create a delegate ?  
   Question 32  :- Where have you used delegates ?  
   Question 33  :- What are Multicast delegates ?  
   Question 34  :- What is an Event ?  
   Question 35  :- How to create an event ?  
   Question 36  :- Delegate vs Events.  
   Question 37  :- What is a class and object ?  
   Question 38  :- Abstraction vs Encapsulation?  
   Question 39  :- Explain Inheritance ?  
   Question 40  :- Explain virtual keyword ?  
   Question 41  :- What is overriding ?  
   Question 42  :- Explain overloading ?  
   Question 43  :- Overloading vs Overriding ?  
   Question 44 :- What is polymorphism ?  
   Question 45 :- Can polymorphism work without inheritance ?  
   Question 46 :- Explain static vs dynamic polymorphism ?  
   Question 47 :- Explain operator overloading ?  
   Question 48  :- Why do we need Abstract classes ?  
   Question 49  :- Are Abstract methods virtual ?  
   Question 50  :- Can we create an instance of Abstract classes ?  
   Question 51  :- Is it compulsory to implement Abstract methods ?  
   Question 52  :- Why does simple base class replace Abstract class ?  
   Question 53  :- Explain interfaces and why do we need it ?  
   Question 54  :- Can we write logic in the interface ?  
   Question 55  :- Can we define methods as private in interface ?  
   Question 56  :- If i want to change the interface what's the best practice ?  
   Question 57  :- Explain Multiple inheritance in Interface ?  
   Question 58  :- Explain Interface Segregation principle ?  
   Question 59  :- Can we create an instance of interface ?  
   Question 60  :- Can we do Multiple inheritance with Abstract classes ?