C#.Net Interview Question and Answers

|  |  |
| --- | --- |
| **1.** | **What is C#?** |
|  | * C# (pronounced "C sharp") is a simple, modern, object-oriented, and type-safe programming language. * It will immediately be familiar to C and C++ programmers. * C# combines the high productivity of Rapid Application Development (RAD) languages. |

|  |  |
| --- | --- |
| **2.** | **What are the types of comment in C#?** |
|  | There are 3 types of comments in C#.   * Single line (//) * Multi (/\* \*/) * Page/XML Comments (///). |

|  |  |
| --- | --- |
| **3.** | **What are the namespaces used in C#.NET?** |
|  | Namespace is a logical grouping of class.   * using System; * using System.Collections.Generic; * using System.Windows.Forms; |

|  |  |
| --- | --- |
| **4.** | **What are the characteristics of C#?** |
|  | There are several characteristics of C# are :   * Simple * Type safe * Flexible * Object oriented * Compatible * Consistent * Interoperable * Modern |

|  |  |
| --- | --- |
| **5.** | **What are the different categories of inheritance?** |
|  | Inheritance in Object Oriented Programming is of four types:   * **Single inheritance** : Contains one base class and one derived class. * **Hierarchical inheritance** : Contains one base class and multiple derived classes of the same base class. * **Multilevel inheritance** : Contains a class derived from a derived class. * **Multiple inheritance** : Contains several base classes and a derived class. |

An abstract class may contain complete or incomplete methods.

Interfaces can contain only the signature of a method but no body.

Thus an abstract class can implement methods but an interface can not implement methods.

An abstract class can contain fields, constructors, or destructors and implement properties.

An interface can not contain fields, constructors, or destructors and it has only the properties signature but no implementation. An abstract class cannot support multiple inheritance, but an interface can support multiple inheritance. Thus a class may inherit several interfaces but only one abstract class. A class implementing an interface has to implement all the methods of the interface, but the same is not required in the case of an abstract Class. Various access modifiers such as abstract, protected, internal, public, virtual, etc. are useful in abstract Classes but not in interfaces

|  |  |
| --- | --- |
| **6.** | **What are the basic concepts of object oriented programming?** |
|  | It is necessary to understand some of the concepts used extensively in object oriented programming.These include   * Objects * Classes * Data abstraction and encapsulation * Inheritance * Polymorphism * Dynamic Binding * Message passing. |

|  |  |
| --- | --- |
| **7.** | **Can you inherit multiple interfaces?** |
|  | Yes. Multiple interfaces may be inherited in C#. |

|  |  |
| --- | --- |
| **8.** | **What is inheritance?** |
|  | Inheritance is deriving the new class from the already existing one. |

|  |  |
| --- | --- |
| **9.** | **Define scope?** |
|  | Scope refers to the region of code in which a variable may be accessed. |

|  |  |
| --- | --- |
| **10.** | **What is the difference between public, static and void?** |
|  | * **public :**The keyword public is an access modifier that tells the C# compiler that the Main method is accessible by anyone. * **static :**The keyword static declares that the Main method is a global one and can be called without creating an instance of the class. The compiler stores the address of the method as the entry point and uses this information to begin execution before any objects are created. * **void :** The keyword void is a type modifier that states that the Main method does not return any value. |