

C# Interview Questions for beginners – Part 1

C# is a general-purpose, object-oriented programming language. The most recent stable version (at the time of writing this post) is **C# 6.0** which was released in 2015. The next version of C# which is **C#7.0** is ready and will be out soon. With the release of [Visual Studio "15" Preview 4](#), most of these (C#7.0) features are coming alive.

Those who want to learn c# from scratch can take this [course](#).

In this post, we have compiled a list of c# interview questions and their answers for beginners as well as experienced users. These questions will help you to prepare for the interviews, for quick revision and provide strength to your technical skills.

1. What is C#?

C# (pronounced as CSharp) is an object-oriented, type safe and managed language that is compiled by .NET Framework to generate Microsoft Intermediate Language (MSIL).

2. What are the types of comment in C# with examples?

- **Single line**

```
1 //This is a Single line comment
```

- **Multiple line (/* */)**

```
1 /*This is a multiple line comment
2 We are in line 2Last line of comment*/
```

- **XML Comments (///)**

```
1 /// These comments can be used to briefly describe a class,
2 /// method, interface or any other entity.
```

3. Can multiple catch blocks be executed?

No. Multiple **catch** blocks can't be executed. Once the proper **catch** code executed, the control is transferred to the **finally** block and then the code that follows the finally block gets executed.

4. What is the difference between public, static and void?

All these are **access/type modifiers** in C#.

public declared variables or methods are accessible anywhere in the application.

static declared variables or methods are globally accessible 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 is a type modifier which states that the method or variable does not return any value.

5. What is an object?

An object is an instance of a class through which we access the methods of that class. **new** keyword is used to create an object. A class that creates an object in memory will contain the information about the methods, variables and behaviour of that class.

6. Define Constructors?

A constructor is a member function in a class that has the same name as its class. The constructor is automatically invoked whenever an object class is created. It constructs the values of data members while initializing the class.

```
1    public class Employee
2    {
3        int empid;
4        string empName;
5
6        public Employee(int a , string e)
7        {
8            empid = a;
9            empName = e;
10       }
11    }
```

7. What are Jagged Arrays?

The array which has elements of type `array` is called jagged array. The elements can be of different dimensions and sizes. We can also call jagged array as **Array of arrays**.

```
1 // Declare the array of two elements:
2 int[][] arr = new int[2][];
3
4 // Initialize the elements:
5 arr[0] = new int[5] { 1, 3, 5, 7, 9 };
6 arr[1] = new int[4] { 2, 4, 6, 8 };
```

8. What is the difference between ref & out parameters?

An argument passed as `ref` must be initialized before passing to the method whereas, `out` parameter needs not to be initialized before passing to a method.

9. What is the use of using statement in C#?

The `using` block is used to obtain a resource and use it and then automatically dispose of when the execution of block completed.

10. What is serialization?

When we want to transport an object through network then we have to convert the object into a stream of bytes. The process of converting an object into a stream of bytes is called **Serialization**. For an object to be serializable, it should inherit `ISerialize` Interface.

De-serialization is the reverse process of creating an object from a stream of bytes.

11. Can 'this' keyword be used within a static method?

We can't use `this` in a static method because we can only use static variables/methods in a static method.

12. What is difference between constants and read-only?

Constant variables are declared and initialized at compile time. The value can't be changed afterwards.

Read-only variables will be initialized only from the Static constructor of the class. Read only is used only when we want to assign the value at run time.

13. What is an interface?

An **interface** is an abstract class which has only **public** abstract methods. The methods only have the declaration and not the definition. These abstract methods must be implemented in the inherited classes.

```
1 public interface IEmployee
2 {
3     public void getEmployeeDetails();
4     public void getEmployeeById(int id);
5 }
```

14. What are value types and reference types?

Value types are stored in the **Stack** whereas, reference types stored on **Heap**.

Value types:

```
1 int, enum, byte, decimal, double, float, long
```

Reference Types:

```
1 string, class, interface, object
```

15. What are Custom Control and User Control?

Custom Controls are controls generated as compiled code (DLLs), those are easier to use and can be added to toolbox. Developers can drag and drop controls to their web forms. Attributes can be set at design time. We can easily add custom controls to Multiple Applications (If Shared DLLs), If they are private then we can copy the DLL to bin directory of web application and then add a reference to use them.

User Controls are very much similar to ASP include files, and are easy to create. User controls can't be placed in the toolbox and dragged – dropped from it. They have their design and code behind. The file extension for user controls is **ascx**. They serve the purpose of re-usability.

16. What are sealed classes in C#?

We create sealed classes when we want to restrict the class to be inherited. **Sealed** modifier used to prevent derivation from a class. If we forcefully specify a sealed class as base class, then a compile-time error occurs.

```
1 sealed class Person
2 {
3     // this is a sealed class and cannot be inherited.
4 }
```

17. What is method overloading?

Method overloading is creating multiple methods with the same name with unique signatures in the same class. When we compile, the compiler uses overload resolution to determine the specific method to be invoked.

Method overloading = Same name, different arguments

18. What is the difference between Array and ArrayList?

In an **array**, we can have items of the same type only. The size of the array is fixed. An **ArrayList** is similar to an **array** but it doesn't have a fixed size.

19. Can a private virtual method be overridden?

No. The reason being private virtual methods are not accessible outside the class.

20. Describe the accessibility modifier 'protected internal'.

protected internal variables/methods are accessible within the same assembly and also from the classes that are derived from this parent class.

21. What are the differences between System.String and System.Text.StringBuilder classes?

System.String is immutable. When we modify the value of a string variable then a new memory is allocated to the new value and the previous memory allocation released. **System.StringBuilder** was designed to have concept of a mutable string where a variety of operations can be performed without allocation separate memory location for the modified string.

For more you can refer – [Difference between String & StringBuilder in C#](#)

22. What's the difference between the System.Array.CopyTo() and System.Array.Clone()?

Using `Clone()` method, we create a new array object containing all the elements in the original array.

With `CopyTo()` method, all the elements of existing array copies into another existing array.

Both the methods perform a shallow copy.

23. How can we sort the elements of the array in descending order?

`Sort()` method followed by `Reverse()` method can help to sort the elements of an array in descending order.

24. Write down the C# syntax to catch exception?

To catch an exception, we use `try - catch` blocks. Catch block can have parameter of `System.Exception` type.

```
1  try
2  {
3      GetDotnetCrunchAuthors();
4  }
5  catch(Exception ex)
6  {
7  }
```

The parameter "ex" from `catch` statement can be skipped as well. It is not mandatory to include it.

25. What's the difference between an interface and abstract class?

Interfaces have all the methods having only declaration but no definition. In an abstract class, we can have some concrete methods.

In an interface, all the methods are `public`. An abstract class may have `private` methods.

Thank you!