**C# Interview Question and Answers**

1. What is C#?

C# is an [object-oriented programming language](https://www.simplilearn.com/tutorials/java-tutorial/oops-interview-questions) compiled by the .Net framework to generate Microsoft Intermediate Language.

C# support Multiple programming language.

Product of Microsoft.

2. Can multiple catch blocks be executed?

No, you cannot execute multiple catch blocks of the same type.

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

Public declared variables can be accessed from anywhere in the application. Static declared variables can be accessed globally without needing to create an instance of the class. Void is a type modifier which states the method and is used to specify the return type of a method in C#.

### 4. What is an object?

An object is a class instance that can be used to access class methods. The "New" keyword can be used to construct an object.

### 5. Define Constructors.

A [constructor](https://www.simplilearn.com/tutorials/c-sharp-tutorial/c-sharp-constructor) is a member function with the same name as its class. The constructor is automatically invoked when an object is created. While the class is being initialized, it constructs all the values of data members.

### 6. What are Jagged Arrays?

The Array which comprises elements of type array is called Jagged Array. The elements in Jagged Arrays can be of various dimensions and sizes.

### 7. What is the difference between out and ref parameters?

When an argument is passed as a ref, it must be initialized before it can be passed to the method. An out parameter, on the other hand, need not to be initialized before passing to a method.

### 8. What is the benefit of ‘using’ statement in C#?

The ‘using’ statement can be used in order to obtain a resource for processing before automatically disposing it when execution is completed.

### 9. What is serialization?

In order to transport an object through a network, we would need to convert it into a stream of bytes. This process is called Serialization.

### 10. Can “this” command be used within a static method?

No. This is because only static variables/methods can be used in a static method.

### 11. Differentiate between Break and Continue Statement.

Continue statement - Used in jumping over a particular iteration and getting into the next iteration of the[loop.](https://www.simplilearn.com/tutorials/asp-dot-net-tutorial/for-each-loop)

Break statement - Used to skip the next statements of the current iteration and come out of the loop.

### 12. Explain the four steps involved in the C# code compilation.

Four steps of code compilation in C# include -

* Source code compilation in managed code.
* Newly created code is clubbed with assembly code.
* The Common Language Runtime (CLR) is loaded.
* Assembly execution is done through CLR.

13. Discuss the various methods to pass parameters in a method.

The various methods of passing parameters in a method include -

* Output parameters: Lets the method return more than one value.
* Value parameters: The formal value copies and stores the value of the actual argument, which enables the manipulation of the formal parameter without affecting the value of the actual parameter.
* Reference parameters: The memory address of the actual parameter is stored in the formal argument, which means any change to the formal parameter would reflect on the actual argument too.

14. Name all the C# access modifiers.

The C# access modifiers are -

* Private Access Modifier - A private attribute or method is one that can only be accessed from within the class.
* Public Access Modifier - When an attribute or method is declared public, it can be accessed from anywhere in the code.
* Internal Access Modifier - When a property or method is defined as internal, it can only be accessible from the current assembly point of that class.
* Protected Access Modifier - When a user declares a method or attribute as protected, it can only be accessed by members of that class and those who inherit it.

### 15. What is meant by Unmanaged or Managed Code?

In simple terms, managed code is code that is executed by the CLR (Common Language Runtime). This means that every application code is totally dependent on the .NET platform and is regarded as overseen in light of it. Code executed by a runtime programme that is not part of the .NET platform is considered unmanaged code. Memory, security, and other activities related to execution will be handled by the application's runtime.

### 16. What is meant by an Abstract Class?

It's a type of class whose objects can't be instantiated, and it's signified by the term 'abstract'. It consists of a methodology or a single approach.

### 17. What is meant by an Interface?

An interface is a class that does not have any implementation. Only the declarations of events, properties, and attributes are included.

### 18. What is meant by a Partial Class?

A [partial class](https://www.simplilearn.com/tutorials/c-sharp-tutorial/partial-class-in-c-sharp) effectively breaks a class's definition into various classes in the same or other source code files. A class definition can be written in numerous files, but it is compiled as a single class at runtime, and when a class is formed, all methods from all source files can be accessed using the same object. The keyword 'partial' denotes this.

### 19. What is the difference between read-only and constants?

During the time of compilation, constant variables are declared as well as initialized. It’s not possible to change this particular value later. On the other hand, read-only is used after a value is assigned at run time.

### 20. What are sealed classes in C#?

When a restriction needs to be placed on the class that needs to be inherited, sealed classes are created. In order to prevent any derivation from a class, a sealed modifier is used. Compile-time error occurs when a sealed class is forcefully specified as a base class.

### 21. What is method overloading?

Method overloading is the process of generating many methods in the same class with the same name but distinct signatures. The compiler utilizes overload resolution to identify which method to invoke when we compile.

### 22. What is the difference between Arraylist and Array?

An array only has items of the same type and its size if fixed. Arraylist is similar but it does not have a fixed size.

### 23. Describe the accessibility modifier “protected internal”.

Variables or methods that are Protected Internal can be accessed within the same assembly as well as from the classes which have been derived from the parent class.

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

System.String is absolute. When a string variable’s value is modified, a new memory is assigned to the new value. The previous memory allocation gets released. System.StringBuilder, on the other hand, is designed so it can have a mutable string in which a plethora of operations can be performed without the need for allocation of a separate memory location for the string that has been modified.

### 25. What’s the difference between the System.Array.CopyTo() and System.Array.Clone() ?

In the Clone() method, a new array object is created, with all the original Array elements using the CopyTo() method. Essentially, all the elements present in the existing array get copied into another existing array.

### 26. How can the Array elements be sorted in descending order?

You can use the Using Sort() methods and then Reverse() method.

### 27. What’s the difference between an abstract and interface class?

All methods in interfaces have only a declaration but no definition. We can have some strong methods in an abstract class. All methods in an interface class are public. Private methods may exist in an abstract class.

### 28. What is the difference between Dispose() and Finalize()methods?

Dispose() is used when an object is required to release any unmanaged resources in it. Finalize(), on the other hand, doesn’t assure the garbage collection of an object even though it is used for the same function.

### 29. What are circular references?

When two or more resources are dependent on each, it causes a lock condition, and the resources become unusable. This is called a circular reference.

### 30. What are generics in C# .NET?

In order to reduce code redundancy, raise type safety, and performance, generics can be used in order to make code classes that can be reused. Collection classes can be created using generics.

### 31. What is an object pool in .NET?

A container that has objects which are ready to be used is known as an object pool. It helps in tracking the object which is currently in use and the total number of objects present in the pool. This brings down the need for creating and re-creating objects.

### 32. What are Custom Exceptions?

In some cases, errors have to be handled according to user requirements. Custom exceptions are used in such cases.

### 33. What are delegates?

Delegates are essentially the same as function [pointers in C++](https://www.simplilearn.com/tutorials/cpp-tutorial/pointers-in-cpp). The main and only difference between the two is delegates are type safe while function pointers are not. Delegates are essential because they allow for the creation of generic type-safe functions.

### 34. What is the difference between method overriding and method overloading?

In method overriding, the relevant method definition is replaced in the derived class, which changes the method behavior. When it comes to method overloading, a method is created with the same name and is in the same class while having different signatures.

### 35. What is the difference between a Struct and a Class?

Structs are essentially value-type variables, whereas classes would be reference types.

### 36. What is the difference between “is” and “as” operators in c#?

An “is” operator can be used to check an object’s compatibility with respect to a given type, and the result is returned as a Boolean. An “as” operator can be used for casting an object to either a type or a class.

### 37. What is a multicast delegate?

Multicast delegate is when a single delegate comes with multiple handlers. Each handler is assigned to a method.

### 38. What are indexers in C# .NET?

In C#, indexers are called smart arrays. Indexers allow class instances to be indexed in the same way as arrays do.

### 39. What is the distinction between "throw" and "throw ex" in.NET?

“Throw” statement keeps the original error stack. But “throw ex” keeps the stack trace from their throw point.

### 40. Difference between SortedList and SortedDictionary in C#.

SortedList is a collection of value pairs sorted by their keys. SortedDictionary is a collection to store the value pairs in the sorted form, in which the sorting is done on the key.

### 41. What is Singleton design pattern in C#?

Singleton design pattern in C# has just one instance that gives global access to it.

### 42. What is tuple in C#?

Tuple is a data structure to represent a data set that has multiple values that could be related to each other.

### 43. What are Events?

An event is a notice that something has occurred.

### 44. What is the Constructor Chaining in C#?

With Constructor Chaining, an overloaded constructor can be called from another constructor. The constructor must belong to the same class.

### 45. What is a multicasting delegate in C#?

Multicasting of delegates helps users to point to more than one method in a single call.

### 46. What are Accessibility Modifiers in C#?

Access Modifiers are terms that specify a program's member, class, or datatype's accessibility.

### 47. What is a Virtual Method in C#?

In the parent class, a virtual method is declared that can be overridden in the child class. We construct a virtual method in the base class using the virtual keyword, and that function is overridden in the derived class with the Override keyword.

### 48. What is Multithreading with .NET?

Multi-threading refers to the use of multiple threads within a single process. Each thread here performs a different function.

### 49. In C#, what is a Hash table class?

The Hash table class represents a collection of key/value pairs that are organized based on the hash code of the key.

### 50. What is the Race condition in C#?

When 2 threads access the same resource and try to change it at the same time, we have a race condition.

### 51. Why are Async and Await used in C#?

Asynchronous programming processes execute independently of the primary or other processes. Asynchronous methods in C# are created using the Async and Await keywords.

### 52. What is an Indexer in C#?

An indexer is a class property that allows you to access a member variable of another class using array characteristics.

### 53. What is Thread Pooling in C#?

In C#, a Thread Pool is a group of threads. These threads are used to do work without interfering with the principal thread's operation.

### 54. What information can you provide regarding the XSD file in C#?

XSD stands for XML Schema Definition. The XML file can have any attributes and elements if there is no XSD file associated with it.

### 55. What are the types of classes in C#?

* **Static class:** Static class, defined by the keyword ‘static’ does not allow inheritance. Therefore, you cannot create an object for a static class.
* **Partial class:** Partial class, defined by the keyword ‘partial’ allows its members to partially divide or share source (.cs) files.
* **Abstract class:** Abstract classes are classes that cannot be instantiated where you cannot create objects. Abstract classes work on the OOPS concept of abstraction. Abstraction helps to extract essential details and hide the unessential ones.
* **Sealed class:** Sealed classes are classes that cannot be inherited. Use the keyword sealed to restrict access to users to inherit that class.

### 56. What are extension methods in C#?

Extension methods help to add new methods to the existing ones. The methods that are added are static. At times, when you want to add methods to an existing class but don’t perceive the right to modify that class or don’t hold the rights, you can create a new static class containing the new methods. Once the extended methods are declared, bind this class with the existing one and see the methods will be added to the existing one.

### 57. What are Generics in C#?

In C# collections, defining any kind of object is termed okay which compromises C#’s basic rule of type-safety. Therefore, generics were included to type-safe the code by allowing re-use of the data processing algorithms. Generics in C# mean not linked to any specific data type. Generics reduce the load of using boxing, unboxing, and typecasting objects.

### 58. What is Boxing and Unboxing in C#?

Boxing : To Convert value type into reference type.

Unboxing: To Convert reference type to value type.

### 59. What are Properties in C#?

Properties in C# are public members of a class where they provide the ability to access private members of a class. The basic principle of encapsulation lets you hide some sensitive properties from the users by making the variables private. The private members are not accessible otherwise in a class. Therefore, by using properties in C# you can easily access the private members and set their values.

### 60. What are partial classes in C#?

Partial classes implement the functionality of a single class into multiple files. These multiple files are combined into one during compile time. The partial class can be created using the partial keyword.

**61. What is Reflection in C#?**

Reflection in C# extracts metadata from the datatypes during runtime.   
  
To add reflection in the .NET framework, simply use System.Refelction namespace in your program to retrieve the type which can be anything from:

* Assembly
* Module
* Enum
* MethodInfo
* ConstructorInfo
* MemberInfo
* ParameterInfo
* Type
* FieldInfo
* EventInfo
* PropertyInfo

### 62. What is the difference between constant and readonly in C#?

A**const** keyword in C# is used to declare a constant field throughout the program. That means once a variable has been declared **const**, its value cannot be changed throughout the program.   
  
In C#, a constant is a number, string, null reference, or boolean values.

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

You can access public declared variables anywhere in the application.

Static declared variables are globally accessible without creating an instance of the class.

Void is a type modifier that specifies that the method doesn't return any value.

### 64. What are the different types of constructors in C#?

* Static constructor
* Private constructor
* Copy constructor
* Default constructor
* Parameterized constructor

### 65. What is static constructor?

Static constructor is used to initialize static data members as soon as the class is referenced first time.

### 66. Is overriding of a function possible in the same class?

No

67. What is array?

Array is a set of related instances either value or reference types.

* **Single Dimensional Array**: It contains a single row. It is also known as vector array.
* **Multi Dimensional Array**: It is rectangular and contains rows and columns.
* **Jagged Array**: It also contains rows and columns but it has an irregular shape.

### 68. What is ArrayList?

ArrayList is a dynamic array. You can add and remove the elements from an ArrayList at runtime. In the ArrayList, elements are not automatically sorted.

### 69. What is a collection?

A collection works as a container for instances of other classes. All classes implement ICollection interface.

### 70. What is serialization?

If you want to transport an object through network then you have to convert the object into a stream of bytes. The process of converting an object into a stream of bytes is called serialization.

71. What is the difference between early binding and late binding in C#?

Early binding and late binding are the concept of polymorphism. There are two types of polymorphism in C#.

* **Compile Time Polymorphism**: It is also known as early binding.
* **Run Time Polymorphism**: It is also known as late binding or method overriding or dynamic polymorphism.

### 72. What is object pool in .Net?

Object pool is a container of ready to use objects. It reduces the overhead of creating new object.

### 73. What is Hashtable?

A Hashtable is a collection of key/value pairs. It contains values based on the key.

### 74. What is Reflection?

Reflection allows us to get metadata and assemblies of an object at runtime.

75.How do you do Exception Handling in C#?

The following four keywords are used for Exception Handling in C#:

* Try - The try block recognizes which block of code has particular exceptions activated.
* Catch - The catch keyword signifies a program for catching an exception using an exception handler.
* Finally - The finally block executes a given block of code whether or not an exception is caught.
* Throw - Using the throw keyword, the program throws an exception in the event of a problem.

76. What is a multicasting delegate?

Multicasting delegates allow users to invoke multiple callbacks. It can refer to multiple methods and functions having the same signature at one time.

77. What are jagged arrays?

A jagged array, also called an array of arrays, is a multidimensional array that consists of other arrays of different sizes.

78. What is a read only variable?

Read only variables are created using the readonly keyword. Its value can be modified only within a constructor.

79. Define nullable types in C#.

Nullable types allow you to assign a normal range to null values. You can also assign true or false to null types/

The syntax is:

< data\_type> ? <variable\_name> = null;

80. Why is finally block used in C#?

The finally block always gets executed if there is an exception or not. When the code is executed in the try block and an exception occurs, control returns to the catch block, and in the end, the finally block gets executed. The finally block therefore can contain closing connections to the database and the release of file handlers.

### 81. What is an object pool in .NET?

An object pool is a container having objects ready to be used. It tracks the object that is currently in use, total number of objects in the pool. This reduces the overhead of creating and re-creating objects.

82. What is the base class in .net from which all the classes are derived from?

System.Object

83. What are C# attributes and its significance?

C# provides developers a way to define declarative tags on certain entities, eg. Class, method, etc. are called attributes. The attribute’s information can be retrieved at runtime using Reflection.

84. How to implement a singleton design pattern in C#?

In a singleton pattern, a class can only have one instance and provides an access point to it globally.

### 85. Explain Deadlock?

A deadlock is a situation that arises when a process isn’t able to complete it’s execution because two or more than two processes are waiting for each other to finish. This usually occurs in multi-threading. In this, a shared resource is being held up by a process and another process is waiting for the first process to get over or release it, and the thread holding the locked item is waiting for another process to complete.

### 86. Illustrate Race Condition?

A Race Condition occurs in a situation when two threads access the same resource and try to change it at the same time. The thread which accesses the resource first cannot be predicted. Let me take a small example where two threads X1 and X2 are trying to access the same shared resource called T. And if both threads try to write the value to T, then the last value written to T will be saved.

### 87. What is Thread Pooling?

A Thread pool is a collection of threads that perform tasks without disturbing the primary thread. Once the task is completed by a thread it returns to the primary thread.

##### **88. What is JIT?**

1. JIT stands for Just-in-time.
2. JIT is the component of CLR that is responsible for converting MSIL code into Native code or Machine code.
3. This Native code or Machine code is directly understandable by the operating system.

##### **89. What is the difference between int.Parse and int.TryParse methods?**

The parse method throws an exception if the string you are trying to parse is not a valid number whereas TryParse returns false and does not throw an exception if parsing fails. Hence TryParse is more efficient than Parse.

##### **90. Can you use virtual, override, or abstract keywords on an accessor of a static property?**

No, it is a compile-time error to use virtual, abstract, or override keywords on an accessor of a static property.

##### **91. If you define a user-defined data type by using the struct keyword, Is it a value type or reference type?**

Value Type

##### **92. If you define a user-defined data type by using the class keyword, Is it a value type or reference type?**

Reference type

##### **93. What is the base class from which all value types are derived?**

System.ValueType

##### **94. Give examples of value types.**

1. Enum
2. Struct

##### **95. Give examples for reference types.**

1. Class
2. Delegate
3. Array
4. Interface

##### **96. If casting fails what type of exception is thrown?**

**InvalidCastException**

##### **97. Is boxing an implicit conversion?**

Yes, boxing happens implicitly.

##### **98. Is unboxing an implicit conversion?**

No, unboxing is an explicit conversion.

##### **99. Can destructors have access modifiers?**

No, destructors cannot have access modifiers.

### **100. Difference between the Equality Operator (==) and Equals() Method in C#?**

### Although both are used to compare two objects by value, still they both are used differently.

### **Equality operator (==) is a reference type which means that if equality operator is used, it will return true only if both the references point to the same object.**

### **Equals() method: Equals method is used to compare the values carried by the objects. int x=10, int y=10. If x==y is compared then, the values carried by x and y are compared which is equal and therefore they return true.**

### **Equality operator: Compares by reference**

### **Equals(): Compares by value**