Home Resources Software Development Top 85 C# Interview Questions and Answers for 2024

Top 85 C# Interview Questions and Answers for 2024

By Ravikiran A S

Last updated on Nov 24, 2023 348553

Share This Article:

C# is a Microsoft-developed, general-purpose, object-oriented programming language that has been accepted by the ISO (International Standards Organization) and ECMA (European Computer Manufacturers Association).

During the development of the .Net Framework, Anders Hejlsberg and his colleagues created C#. C# is intended for the CLI (Common Language Infrastructure), consisting of a runtime environment and an executable code that enables the use of a variety of high-level languages on a variety of computer platforms and architectures.

C# programming features:

- Windows Integration
- A library that is standard

- Automated Garbage Collection
- Boolean Conditions
- Lambda Expressions and LINQ
- Proper Indexers
- Generics that are easy-to-use
- Assembly Versioning
- Multithreading that is simple
- Event Managers and Delegates
- Properties and Events
- Conditional Compilation

Learning Roadmap (Tools and Skills Required)

To become proficient in C#, you need to master a few other core concepts and design principles that are prerequisites to learning C#. Having a deep knowledge of these subjects will enable you learn C# quickly and efficiently, and you will be able to write clear, maintainable code.

Prerequisites to Learning C#:

Basic knowledge of computers, such as working with command-line, file systems, and installing applications such as compilers, interpreters, etc.

You will need to have some understanding of concepts like encapsulation, polymorphism, abstraction, inheritance, interfaces, etc., as C# is an object-oriented programming language.

Basic knowledge of C, C++, or Java so you can understand the syntax of C# better.

C# is one of the most flexible programming languages with a powerful runtime, access to a huge amount of library and platform functionality, and a strong set of abstractions.

1. Syntaxes, Variables, and Data Types

The basic step for learning any language is understanding the syntax. It is the same with C#. You will need to learn and understand how to declare variables. You will also need to understand all the various data types, type-conversion and comments, among others.

2. Conditionals and Loops

Once you understand the syntax, your next step would be to learn about branching and looping statements. Loops, in particular, help you in creating programs that require repetitive tasks. This way, you can avoid running the code several times manually. Conditionals such as If/Else conditions can help in giving a flow to your program.

3. Functions/Methods and Arrays

Functions and methods allow code to be grouped together in order to perform specific functions like adding two numbers. If you need a specific operation to be performed multiple times, you can create a simple function and use it as needed. Also, whenever you need to store data, arrays will help you.

4. Classes and Objects

Classes and objects are part of OOPS. Classes capture all the related data together, while objects can help provide the instances in which to gain access to the information stored inside classes.

5. Interfaces and Inheritance

Inheritance can help inherit functions and variables of one or more classes into another. Interfaces, on the other hand, will let you create classes in the form of structures in which you can just declare the methods and not have to define them. When required, you can define the methods in subclasses at a later time.

Opportunities and Average Salary

C# developers are usually part of a workgroup made up of various disciplines and employed to build and support server-client applications as well as web-based applications that use programs such as C#, ASP.NET framework and SQL, among others. Generally, C# developers play a part in the complete development cycle of an application. C# developers would also be expected to maintain applications that were implemented by previous developers.

As such, there are a lot of opportunities for C# developers in the job market. On average, a C# developers can expect to make approximately $84,243 annually. There are various roles such as Software Engineer, Senior Software Developer and Software Programmer that candidates can apply for.

C# Interview Questions

Now, let us take a look at the top C# interview questions that you might face!

1. What is C#?

C# is an object-oriented programming language compiled by the .Net framework to generate Microsoft Intermediate Language.

Can multiple catch blocks be executed?

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

2. 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#.

3. 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.

4. Define Constructors.

A 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.

5. 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.

6. 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.

7. 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.

8. 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.

9. Can "this" command be used within a static method?

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

10. Differentiate between Break and Continue Statement.

Continue statement - Used in jumping over a particular iteration and getting into the next iteration of the loop.

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

11. List the different types of comments in C#.

The different types of comments in C# are:

- XML comments

Example -

/// example of XML comment

- Single Line comments

Example -

// example of single-line comment

- Multi-line comments

Example -

/* example of an

multiline comment */

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. Mention all the advantages of C#.

The following are the advantages of C# -

- C# is component-oriented.

- It is an object-oriented language.
- The syntax is really easy to grasp.
- It is easier to learn.
- C# is part of the framework called .NET

16. Mention the important IDEs for C# development provided by Microsoft.

The following IDEs' are useful in C# development -

- MonoDevelop
- Visual Studio Code (VS Code)
- Browxy
- Visual Studio Express (VSE)
- Visual Web Developer (VWD)

17. Why do we use C# language?

Below are the reasons why we use the C# language -

- C# is a component-oriented language.
- It is easy to pass parameters in the C# language.

- The C# language can be compiled on many platforms.
- The C# language follows a structured approach.
- It is easy to learn and pick up.
- The C# language produces really efficient and readable programmes.

18. Mention the features of C# briefly.

Some of the main features of C# are -

- C# is a safely typed and managed language.
- C# is object-oriented in nature.
- C# is a Cross-platform friendly language.
- C# is a platform-independent language when it comes to compilation.
- C# is general purpose in nature.
- C# is used in implementing Destructors and Constructors.
- C# is part of the .NET framework.
- C# is an easy-to-learn and easy-to-grasp language.
- C# is a structured language.

19. 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.

20. 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'. consists of a methodology or a single approach.

21. Differentiate between finalize blocks and finalize.

Once the try and catch blocks have been completed, the finalize block is called since it is used for exception handling. No matter if the exception has been captured, this block of code is run. In general, the code in this block is cleaner.

Just before garbage collection, the finalize method is called. The main priorities of the finalize method are to clean up unmanaged code, which is automatically triggered whenever an instance is not re-called.

22. 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.

23. What is meant by a Partial Class?

A partial class 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.

24. 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 p[illegible]le to change this particular value later. On the other hand, read-only is used after a value is assigned at run time.

run time.

25. What is an interface class?

An interface class is an abstract class with only public abstract methods. Only declaration is there in these methods, but not the definition. They must be implemented in the inherited classes.

26. What are reference types and value types?

A value type holds a data value inside its memory space. Reference type, on the other hand, keeps the object's address where the value is stored. It is, essentially, a pointer to a different memory location.

27. What are User Control and Custom Control?

Custom Controls are produced as compiled code. These are easy to use and can be added to the toolbox. Developers can drag and drop these controls onto their web forms. User Controls are almost the same as ASP include files. They are also easy to create. User controls, however, can't be put in the toolbox. They also can't be dragged and dropped from it.

28. 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.

29. 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.

30. 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.

31. Is it possible for a private virtual method to be overridden?

A private virtual method cannot be overridden as it can’t be accessed outside the class.

32. 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.

33. 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.

34. 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.

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

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

36. 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.

37. 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.

38. 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.

39. 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.

40. 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.

41. List down the most commonly used types of exceptions in .NET

Commonly used types of exceptions in .NET are:

ArgumentException

ArithmeticException

DivideByZeroException

OverflowException

InvalidCastException

InvalidOperationException

NullReferenceException

OutOfMemoryException

StackOverflowException

42. What are Custom Exceptions?

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

43. What are delegates?

Delegates are essentially the same as function pointers in C++. 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.

44. 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.

45. How do you inherit a class into another class in C#?

In C#, colon can be used as an inheritance operator. You need to place a colon and follow it with the class name.

46. What are the various ways that a method can be overloaded??

Different data types can be used for a parameter in order for a method to be overloaded; different

Different data types can be used for a parameter in order for a method to be overloaded, different orders of parameters as well as different numbers of parameters can be used.

47. Why can't the accessibility modifier be specified for methods within the interface?

In an interface, there are virtual methods which do not come with method definition. All the methods present are to be overridden in the derived class. This is the reason they are all public.

48. How can we set the class to be inherited, but prevent the method from being overridden?

To set the class to be inherited, it needs to be declared as public. The method needs to be sealed to prevent any overrides.

49. What happens if the method names in the inherited interfaces conflict?

A problem could arise when the methods from various interfaces expect different data. But when it comes to the compiler itself, there shouldn’t be an issue.

50. What is the difference between a Struct and a Class?

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

51. How to use nullable types in .Net?

When either normal values or a null value can be taken by value types, they are called nullable types.

52. How can we make an array with non-standard values?

An array with non-default values can be created using Enumerable.Repeat.

53. 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.

54. What is a multicast delegate?

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

55. 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.

56. 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.

57. What are C# attributes and its significance?

C# gives developers an option to define declarative tags on a few entities. For instance, class and method are known as attributes. The information related to the attribute can be retrieved during runtime by taking the help of Reflection.

58. In C#, how do you implement the singleton design pattern?

In a singleton pattern, a class is allowed to have only one instance, and an access point is provided to it globally.

59. What's the distinction between directcast and ctype?

If an object is required to have the run-time type similar to a different object, then DirectCast is used to convert it. When the conversion is between the expression as well as the type, then Ctype is used.

60. Is C# code managed or unmanaged code?

C# is a managed code as the runtime of Common language can compile C# code to Intermediate language.

61. What is a Console application?

An application that is able to run in the command prompt window is called a console application.

62. What are namespaces in C#?

Namespaces allow you to keep one set of names that is different from others. A great advantage of namespace is that class names declared in one namespace don’t clash with those declared in another namespace.

63. What is the distinction between the Dispose() and Finalize() methods?

Namespaces, interfaces, structures, and delegates can all be members.

64. Write features of Generics in C#?

Generics is a technique to improve your program in various ways including creating generic classes and reusing code.

65. 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.

66. What is Singleton design pattern in C#?

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

67. 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.

68. What are Events?

An event is a notice that something has occurred.

69. 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.

70. What is a multicasting delegate in C#?

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

71. What are Accessibility Modifiers in C#?

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

72. 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.

73. 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.

74. 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.

75. What is LINQ in C#?

LINQ refers to Language Integrated Query. It provides .NET languages (like C#) the ability to generate queries to retrieve data from the data source.

76. Why can't a private virtual procedure in C# be overridden?

Private virtual methods are not accessible outside of the class.

77. What is File Handling in C#?

File handling includes operations such as creating the file, reading from the file, and appending the file, among others.

78. What do you understand about Get and Set Accessor properties?

In C#, Get and Set are termed accessors because they use properties. Such private fields are accessed via accessors.

79. 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.

80. 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.

81. 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.

82. 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.

83. 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.

84. What are I/O classes in C#?

In C#, the System.IO namespace contains multiple classes that are used to conduct different file operations such as creation, deletion, closure, and opening.

85. What exactly do you mean by regular expressions in C#?

A regular expression is a pattern that can be used to match a set of input. Constructs, character literals, and operators are all possible.

Choose The Right Software Development Program

This table compares various courses offered by Simplilearn, based on several key features and details. The table provides an overview of the courses' duration, skills you will learn, additional benefits, among other important factors, to help learners make an informed decision about which course best suits their needs.

Program Name	Full Stack Java Developer Career Bootcamp	Automation Testing Masters Pr
Geo	IN	All
University	Simplilearn	Simplilearn
Course Duration	11 Months	11 Months

Coding Experience Required	Basic Knowledge	Basic Knowledge
Skills You Will Learn	15+ Skills Including Core Java, SQL, AWS, ReactJS, etc.	Java, AWS, API Testing, TDD
Additional Benefits	Interview Preparation Exclusive Job Portal 200+ Hiring Partners	Structured Guidance Learn From Experts Hands-on Training
Cost	$$	$$
	Explore Program	Explore Program

Master C Programming With Simplilearn!

C# is the primary programming language which helps in every higher level programming language. C# developers are in high demand and can find lucrative job opportunities in fields like data science, front end development, backend development, or app development. And since Microsoft backs it, the popularity of C# will continue to increase.

Learning C# can really help accelerate your career. You can enroll in the Full Stack Web Developer Mean Stack course by Simplilearn, which provides full understanding of software development and testing technologies such as JavaScript, Node.js, Angular, Docker, and Protractor You'll create an end-to-end application, test and deploy code, and use MongoDB to store data.

Stack developers who have completed the Full Stack Web Developer Mean Stack course find good job prospects and salary hikes over a period of time.

Check out Simplilearn's Cloud Architect Certification Course, which will build your AWS, Microsoft Azure and Google Cloud Platform expertise from the ground up. You'll learn to master the architectural principles and services of the top cloud platforms, design and deploy highly scalable, fault-tolerant applications and develop skills to transform yourself into an AWS, Azure cloud and

GCP architect.

After completion of this cloud architect certification, you become eligible for some of the rewarding career opportunities as a Solutions Architect, Network Architect, Software Architect, and Cloud Engineer. Enroll today!

Our Software Development Courses Duration And Fees

Software Development Course typically range from a few weeks to several months, with fees varying based on program and institution.

Professional Certificate Program in Blockchain	Cohort Starts: **15 Feb, 2024**	4 Months	**$ 3,000**
Post Graduate Program in Full Stack Web Development		9 Months	**$ 2,000**
Automation Test Engineer	Cohort Starts: **14 Feb, 2024**	11 Months	**$ 1,299**
Full Stack Developer - MEAN Stack	Cohort Starts: **16 Feb, 2024**	11 Months	**$ 1,299**
Full Stack Java Developer	Cohort Starts: **19 Mar, 2024**	6 Months	**$ 1,500**

Recommended Reads

Kubernetes Interview Guide

15 May, 2020

SOFTWARE DEVELOPMENT

Top 30 C# OOPS Interview Questions and Answers for 2022

51307

21 Jul, 2023

SOFTWARE DEVELOPMENT

Top 60 C++ Interview Questions and Answers for 2023

681319

18 Aug, 2023

Disclaimer

PMP, PMI, PMBOK, CAPM, PgMP, PfMP, ACP, PBA, RMP, SP, and OPM3 are registered marks of the Project Management Institute, Inc.