

## Interview Questions

I have come up with these questions after preparing for 3 months and attending a dozen of interviews. I would request you to go ahead and try to understand the answer for each question (which would surely pop up more questions in mind. Find answers for those also and try to fully understand the concept).

After attending interviews ranging from small startups to BIG 4s to Finance Companies to top Product companies in the World, I have realized that what an interviewer always looks is for candidate to make sense. Questions are asked not to judge how well one remembers facts and theories but to check the logical reasoning of mind. If you can provide 2-3 solutions for a problem (maybe not all are correct), while making sense, you will be considered a good candidate for the job. Interviews are majorly focused on to check the thought process of the candidate.

Tips:

1. Always write down the questions which you could not answer and later look for their answers.
2. Keep talking about what you are thinking. Do not be shy.

Contact Email: [12bce1006@gmail.com](mailto:12bce1006@gmail.com)

*Happy Learning* 😊

### 1. OOPs

1. What constitutes a class? What is an object?
2. What is the use of abstraction? Explain with a sample code.
3. What is the difference between abstract and base class? When would you choose a base class over abstract class and vice-versa?
4. What are virtual methods and abstract methods?
5. What is encapsulation? Explain auto-implemented properties and what role do they play in encapsulation?
6. What is polymorphism? What is the difference between method overloading, method overriding and shadowing? Create objects and showcase each technique.
7. What is Inheritance? What are different levels or types of inheritance? What inheritances helps us to achieve?
8. Why would you assign a base class reference to derived class object?
9. Why is it not possible to assign a derived class reference to base class object?
10. When to use a static keyword? What is the use of static class? Static vs Abstract class.
11. Explain the difference between “AS A”, “IS A” and “HAS A” relationships?
12. Explain SOLID Principles with code examples. Mention scenarios when they are useful. How these principles help to write scalable, reusable and cleaner code?
13. What is an interface? Why interface do not have access modifiers? Difference between Abstract class and interface?
14. How interface improves separation of concerns and loose coupling between modules?
15. What do you understand by “Increase Cohesion, Decrease coupling” in Software Architecture Design?
16. Explain Dependency Inversion Principle? Explain with code example.

17. What are constructors? When would you create a Private constructor?
18. How constructors are used to pass parameters in Inheritance?
19. Explain Singleton Design Pattern? What is the use of Singleton class?
20. What is exception handling? (Try, Catch, Finally) Order/Sequence in catching exception
21. What is the difference between deep copy and shallow copy?
22. What is the difference between value and reference type? Is string an entirely reference type?
23. What are access modifiers? Explain Public, Protected, Private, Internal, Protected Internal
24. Explain following keywords and their use in C# with code samples
  1. Sealed
  2. Static
  3. Const
  4. Read Only
25. Difference between Const and Read Only. When would you choose one over other?
26. Class Child: Class Parent (Child class inherits parent class). Which statement is correct and why?
  - i. *Child C = new Parent ();*
  - ii. *Parent P = new Child ();*

## 2. C# and .NET

1. What is .NET Framework?
2. What are the latest features of C# language?
3. How the code gets compiled in .NET Framework? CLR, CTS, MSIL
4. What is GAC? (Global Assembly Cache)
5. What is garbage collection? Different generations in Garbage Collection? Mark and Sweep algorithm.
6. What is the difference between Finalize and Dispose?
7. What is the difference between a dll and exe file?
8. What are the data structures used in C#?
9. What are collections? Explain Array, List, Dictionary, Enumerable. When to use which one?
10. What are Generics? Write sample code snippets to show generic classes, methods and properties? What is the advantage of using Generics?
11. Explain some best practices that one would follow while writing C# code and why?
12. What is Task Parallel Library? (TPL)
13. Explain the difference between concurrency and parallelism? (Single vs Multiple Core)
14. Explain asynchronous programming and provide one use-case where it can be helpful?
15. How would you convert a C# project to a NuGet library?
16. What is role of Web.config and App.config in a .NET project?
17. What are the features of .NET core?
18. How would you compare two objects in C#? Explain the difference between Objects.Equals and ==. (Value vs Reference comparison)
19. What is a Delegate? Write a sample code to show how delegates work.
20. What is the difference between ref and out keyword? (two-way vs one-way binding)
21. What is the use of writing an integration and unit test?
22. What library would you use to unit test a C# method?
23. While adding dependency injection to your code, how do you manage the lifetime of objects? (Hint: Transient/Scope/Singleton)

### 3. Entity Framework and LINQ

1. What is Entity Framework? How is it different from ADO.NET?
2. What are advantages and disadvantages of using Entity Framework?
3. Explain various approaches in Entity Framework? (Mode First, Code First and Database First)
4. How would you create a One-One FK relationship between two entities?
5. How would you create a One-Many FK relationship between two entities?
6. Write some data annotations used in Entity Framework.
7. What is DbContext and DbSet?
8. What is repository pattern in C#?
9. What is LINQ? Define Method Syntax and Query syntax in LINQ.
10. Write a LINQ query to join two tables and perform basic query operations (sorting, grouping)
11. What is the difference between Lazy loading and eager loading?
12. When would you choose IQueryable over IEnumerable?
13. Does LINQ have any performance issues over large set of data?
14. When does the LINQ query executes on Db server?
15. What is Dapper? (wrapper over ADO.NET) When would you choose to use Dapper over Entity Framework?

### 4. SQL

1. What is SQL? What is a database? What is the use of RDBMS?
2. Write small SQL statements showing use of SET Functions and Group by clause.
3. What is Normalization in SQL? Provide example of each form in Normalization.
4. Difference between DDL and DML operations?
5. What are different types of Joins? When would you use a Self-Join?
6. What is a Primary Key, Foreign Key, Composite Key?
7. What are triggers? Explain Before and After triggers.
8. Explain the difference between Views, Functions and Stored Procedures?
9. When would you use a Stored Procedure over a function?
10. Why Stored procedure is faster than set a SQL statements?
11. What is User Defined Function in SQL? Scalar and Table-Valued functions
12. What are indexes? Why do we use them? What is the difference between Clustered and Non-cluster index? (Unique vs Non-unique)
13. What are covering and filtered indexes?
14. What is the disadvantage of using indexes?
15. How would you improve a performance of an SQL Query?
16. What is Query Execution Plan? How do you read it?
17. How would you improve a performance of Stored Procedure? Does the order of joins matter?
18. What are best practices for naming Views, Stored Procedures, Indexes and Primary Keys?
19. Why can we only have one clustered index but multiple non-clustered indexes on a table?
20. What is SQL Deadlock?
21. If multiple updates are performed to same row in SQL simultaneously, how does SQL handles such operations? In other words, how SQL handles concurrent operations?
22. Write and explain some Query and Table Hints in SQL.
23. What is the difference between RANK, ROW\_NUM and DENSE\_RANK?
24. What are different SQL constraints?
25. Explain the difference between TRUNCATE and DELETE.

26. How would you commit a transaction in SQL? How to roll back the same?

#### 5. Miscellaneous (Web API, Software Engineering, DevOps)

1. What is a REST API? What is the difference between REST and SOAP?
2. What protocols are widely used when creating a Web API or Web Service?
3. What is the difference between PUT and POST methods?
4. How do you pass the data to Web API through the body of the request?
5. After deploying a Web API or a Web Service, why the first response is slower than following response?
6. How would you perform data validation in Web API?
7. What is attribute-routing? How would you pass multiple parameters to a Web API in query string?
8. What are filters in Web API? How would you go creating a custom action filter which logs all request and response in a text file?
9. What are the best practices to follow while writing a Web API and why?
10. What are various tools available to test Web API?
11. How would you improve a performance of Web Application? If your web API is taking longer to respond, how would go debugging the issue in both Backend and Database side?
12. What is role of controller in Web API? What are it's features?
13. Where do you host a Web Application? What is IIS? What is Application pool?
14. Can we have multiple applications running under same app pool?
15. How IIS helps in adding security and containerization of a web application?
16. When would you create a Windows Service rather than Web Service or Web API?
17. How do you debug a Windows Service? (2-3 ways in which it can be done)
18. What is the difference between Authentication and Authorization?
19. How do you add Authorization to a Web Application? Explain different ways.
20. What do you understand by DevOps? What is CI and CD?
21. What is the difference between Build and Release Pipeline? What features should each pipeline include to build a more effective DevOps culture?
22. What is a version control system? Explain Push, Pull, Commit, Stage, Fetch, Merge.
23. What branching model would you prefer for developing and deploying a software application?
24. What is agile methodology? Does it really help to be an agile team? If yes, then how?
25. What is Test Driven Development? What do you understand by having cross-functional team to support DevOps?
26. Explain the architecture of your current project. Do you have any downstream and upstream systems? How do they integrate to your current application?
27. What tasks do you believe can be automated in your current project?

*P.S. I have not included any of the UI questions as technology keeps changing, if you are working on Angular or React or any latest UI/UX language, just google search interview questions about the same and those will be good enough for interviews.*