# 💻 C# Interview Questions and Answers (1–50)

## 1. What is C#?

**Answer:** C# is a modern, object-oriented programming language developed by Microsoft as part of the .NET platform.

## 2. What are the key features of C#?

**Answer:**

- Object-oriented
- Strong type checking
- Automatic garbage collection
- Properties and events
- LINQ support

## 3. What is the .NET framework?

**Answer:** A software development platform for building and running applications on Windows.

## 4. What is the CLR?

**Answer:** Common Language Runtime — it manages code execution, memory, security, and more.

## 5. What is the difference between value and reference types?

**Answer:**

- Value types store data directly
- Reference types store references to the data

## 6. What are value types in C#?

**Answer:** `int`, `char`, `float`, `bool`, `struct`, `enum`

## 7. What are reference types in C#?

**Answer:** `class`, `interface`, `delegate`, `string`, `object`

## 8. What is boxing and unboxing?

**Answer:**

- Boxing: Converting value type to object
- Unboxing: Extracting value type from object

### 9. What is a constructor?

**Answer:** A special method used to initialize an object.

### 10. What is a destructor?

**Answer:** A method that is automatically invoked when an object is destroyed.

### 11. What is method overloading?

**Answer:** Defining multiple methods with the same name but different parameters.

### 12. What is method overriding?

**Answer:** Redefining a base class method in a derived class using `override` keyword.

### 13. What is a static class?

**Answer:** A class that cannot be instantiated. All members must be static.

### 14. What is the difference between `const`, `readonly`, and `static`?

**Answer:**

- `const`: Compile-time constant
- `readonly`: Runtime constant
- `static`: Shared among all instances

### 15. What is inheritance in C#?

**Answer:** Enables a class to inherit members of another class.

### 16. What are access modifiers in C#?

**Answer:** `public`, `private`, `protected`, `internal`, `protected internal`, `private protected`

### 17. What is abstraction?

**Answer:** Hiding complex implementation details and showing only essential features.

### 18. What is encapsulation?

**Answer:** Binding data and methods into a single unit.

### 19. What is polymorphism?

**Answer:** Ability to present the same interface for different data types.

## 20. What is interface in C#?

**Answer:** A contract that defines a set of methods/properties without implementation.

## 21. What is the difference between an abstract class and an interface?

**Answer:**

- Abstract class can have implementation
- Interface can’t (till C# 8.0 default methods)

## 22. What are sealed classes?

**Answer:** Classes that cannot be inherited.

## 23. What is a delegate?

**Answer:** A type-safe function pointer.

## 24. What is an event in C#?

**Answer:** A message sent by an object to signal the occurrence of an action.

## 25. What is a lambda expression?

**Answer:** An anonymous function used to create delegates or expression tree types.

## 26. What is LINQ?

**Answer:** Language Integrated Query — used to query collections.

## 27. What are generics in C#?

**Answer:** Allow classes and methods to operate on any data type.

## 28. What is the difference between == and `.Equals()` in C#?

**Answer:**

- `==`: Compares references for reference types
- `.Equals()`: Compares values

## 29. What is an assembly?

**Answer:** A compiled code library used for deployment, versioning, and security.

## 30. What is a namespace?

**Answer:** Organizes classes and other types under a name.

## 31. What is the difference between Array and ArrayList?

**Answer:**

- Array: Fixed size, type-safe
- ArrayList: Dynamic, not type-safe (use `List<T>` instead)

## 32. What is exception handling?

**Answer:** Mechanism to handle runtime errors using `try-catch-finally`.

## 33. What is `finally` block?

**Answer:** Executes whether or not an exception is thrown.

## 34. What is the `using` statement?

**Answer:** Ensures the disposal of IDisposable objects like file or database connections.

## 35. What is multithreading?

**Answer:** Enables concurrent execution of two or more threads.

## 36. What is async/await in C#?

**Answer:** Used for asynchronous programming to keep UI responsive.

## 37. What is a nullable type?

**Answer:** Allows a value type to be assigned null, using `int?`, `bool?`, etc.

## 38. What is a var keyword?

**Answer:** Used for implicit typing; compiler infers the type.

## 39. What is a dynamic keyword?

**Answer:** Bypasses compile-time type checking.

## 40. What are indexers?

**Answer:** Allow objects to be indexed like arrays.

## 41. What are properties in C#?

**Answer:** Encapsulated methods to get/set values.

## 42. What is difference between `ref` and `out`?

**Answer:**

- `ref`: Variable must be initialized before passing
- `out`: Variable is initialized in the called method

## 43. What is extension method?

**Answer:** Adds new methods to existing types without modifying them.

## 44. What is method hiding?

**Answer:** Use of `new` keyword to hide base class method in derived class.

## 45. What is `is` and `as` operators?

**Answer:**

- `is`: Checks object type
- `as`: Attempts to cast

## 46. What is a partial class?

**Answer:** Class definition can be split across multiple files.

## 47. What is a nullable operator `??`?

**Answer:** Returns the left-hand operand if not null; otherwise, returns right-hand.

## 48. What is the difference between `Task` and `Thread`?

**Answer:**

- `Task`: Higher-level abstraction for concurrency
- `Thread`: Lower-level, manual thread management

## 49. What is difference between compile time and run time polymorphism?

**Answer:**

- Compile time: Method overloading
- Run time: Method overriding

## 50. What is garbage collection in C#?

**Answer:** Automatic memory management that reclaims unused objects.