30 C# Interview Question

---

1. What is C#?

C# is a modern, object-oriented, and type-safe programming language developed by Microsoft as part of the .NET framework. It's used to build a variety of applications including web, desktop, mobile, and games.

---

2. What are the main features of C#?

- Object-Oriented
- Type-Safe
- Automatic Garbage Collection
- Rich Standard Library

- Versioning Support
- Interoperability

---

3. What is the difference between ref and out parameters in C#?

- ref requires that the variable be initialized before passing.
- out does not require initialization before passing but must be assigned inside the method.

---

4. What is the difference between == and .Equals()?

- == checks for **reference equality** for reference types.
- .Equals() checks for **value equality** and can be overridden.

---

5. What are value types and reference types in C#?

- **Value types**: stored in stack (e.g., int, float, bool, struct)
- **Reference types**: stored in heap (e.g., class, array, string)

---

6. What is boxing and unboxing?

- **Boxing**: Converting a value type to object type.
- **Unboxing**: Converting object type back to value type.

---

7. What is the difference between abstract class and interface?

- Abstract class: Can have implementations; supports constructors.

- Interface: Only method signatures; multiple interfaces can be implemented.

---

8. What is the difference between const, readonly, and static?

- const: Compile-time constant.
- readonly: Runtime constant, assigned in constructor.
- static: Belongs to the class, not instance.

---

9. What is inheritance in C#?

Inheritance allows one class (child) to inherit fields, methods, and properties from another class (parent).

---

10. What is polymorphism?

Polymorphism allows methods to behave differently based on the object that is calling them. Achieved using **method overloading** and **method overriding**.

---

11. What is encapsulation?

Encapsulation is the concept of hiding internal details and exposing only necessary components through access modifiers.

---

12. What is the purpose of the using statement in C#?

It ensures that IDisposable objects like file streams are disposed of properly, even if an exception occurs.

---

13. What is the difference between throw and throw ex?

- throw: Preserves the original stack trace.
- throw ex: Resets the stack trace.

---

14. What is the difference between Task, async, and await?

- Task: Represents an asynchronous operation.
- async: Marks a method as asynchronous.
- await: Waits for the task to complete without blocking the thread.

---

15. What is the difference between var, dynamic, and object?

- var: Compile-time type inferred.
- dynamic: Type resolved at runtime.
- object: Base type of all types.

---

16. What are access modifiers in C#?

- public, private, protected, internal, protected internal, private protected.

---

17. What is LINQ?

Language Integrated Query (LINQ) is used to query collections in a consistent manner using C# syntax.

---

18. What is the difference between IEnumerable and IQueryable?

- **IEnumerable:** Executes in memory, used for in-memory collections.
- **IQueryable:** Supports querying against a database (deferred execution).

---

19. What is the difference between override, new, and virtual?

- virtual: Base method that can be overridden.
- override: Derived method that overrides the base.
- new: Hides the base class method.

---

20. What is a delegate in C#?

A delegate is a type-safe function pointer that can reference a method with a specific signature.

---

21. What is an event in C#?

An event is a way for a class to notify other classes when something happens, using delegates.

---

22. What is a nullable type?

Allows value types (like int) to hold null values using int?.

---

23. What is the difference between Dispose() and Finalize()?

- Dispose(): Explicitly frees resources.
- Finalize(): Called by GC, not deterministic.

---

24. What is a struct in C#?

A lightweight value type that doesn't support inheritance but can implement interfaces.

---

25. What is a static class?

A class that cannot be instantiated and contains only static members.

26. What is dependency injection?

A design pattern where dependencies are provided rather than created within the class, improving testability and maintainability.

---

27. What is the params keyword?

Allows a method to accept a variable number of arguments as an array.

---

28. What is a lambda expression?

A concise way to represent anonymous methods using =>.

Example: (x, y) => x + y

---

29. What is a tuple in C#?

A data structure to hold a set of elements (can be different types).

Example: (int, string) person = (1, "John");

---

30. What is async/await and how does it work?

It allows asynchronous programming where async marks a method for async operations and await pauses execution until the awaited task completes.
