

1) Introduce Yourself

Good morning/afternoon, my name is [Your Name]. I have completed [Your Degree/Training] and have hands-on experience in [Technology/Domain]. I am passionate about learning new technologies and solving real-world problems. My strengths are problem-solving and quick learning.

2) Object and Class

Class is a blueprint of an object. Object is an instance of a class. Syntax: `class Student { public string name; } Student obj = new Student();`

3) What is () in new User()

The () calls the constructor of the class User when creating an object.

4) User() is which type of constructor

It is a default or parameterized constructor depending on its definition.

5) Types of Constructors

1) Default Constructor 2) Parameterized Constructor 3) Copy Constructor 4) Static Constructor 5) Private Constructor

6) Annotation in C#

In C#, they are called Attributes. They add metadata to code elements like classes, methods.

7) Example of Annotation

`[Obsolete("This method is deprecated")]`

8) CLR

Common Language Runtime - It is the runtime environment of .NET that executes code, does memory management, exception handling, garbage collection.

9) Four Pillars of OOP

1) Encapsulation 2) Abstraction 3) Inheritance 4) Polymorphism

10) Polymorphism

Polymorphism means one name, many forms — the ability to perform a single action in different ways.

11) Types of Polymorphism

1) Compile Time (Method Overloading) 2) Runtime (Method Overriding)

12) Runtime vs Compile Time Polymorphism

Compile-time: Resolved at compile time (Overloading) Runtime: Resolved at runtime (Overriding)

13) Encapsulation

Wrapping data and methods together. Achieved using access modifiers and properties.

14) Abstraction

Hiding implementation details and showing only essential features. Achieved using abstract classes and interfaces.

15) Difference between Encapsulation and Abstraction

Encapsulation → Hides data using access modifiers. Abstraction → Hides implementation using abstract/interface.

16) Can we create reference of Abstract Class?

Yes, we can create a reference but not an object.

17) Interface

Interface is a contract that contains only definitions, no implementation. Used to achieve multiple inheritance.

18) Difference between Abstract and Interface

Abstract: Can have abstract + concrete methods, fields. Interface: Only method signatures and default properties.

19) Can we create reference of Interface?

Yes, reference can be created but object cannot.

20) Access Modifier of Interface

Always public by default.

21) Abstract vs Interface Based on Inheritance

Class can inherit only one abstract class but can implement multiple interfaces.

22) Types of Inheritance

1) Single 2) Multi-level 3) Hierarchical 4) Multiple (via interface) 5) Hybrid

23) Types of Data Types

1) Value Types 2) Reference Types

24) Primitive vs Non-Primitive

Primitive: int, char, float Non-Primitive: class, array, string

25) User Defined Datatype

Yes, we can create class, struct, enum.

26) Null value to int?

No, but we can use nullable int → int? x = null;

27) Value Type vs Reference Type

Value type stored in stack, holds data. Reference type stored in heap, holds reference.

28) Sealed Keyword

Used to prevent inheritance of a class.

29) Partial Class

Allows class definition to be split across multiple files.

30) Static

Static belongs to class not object.

31) Static Class

Yes, we can create static class but cannot create object of it.

32) Non-static method in Static Class?

No, all members of static class must be static.

33) Const

Const is a compile-time constant value.

34) Const vs Static

Const → Value fixed at compile time. Static → Value can change at runtime.

35) Changing value of Static and Const

Static can be changed, Const cannot.

36) Garbage Collection

Process of automatic memory management to free unused objects.

37) Primary Key vs Unique Key

Primary Key: Unique + Not Null. Unique Key: Unique but allows one null.

38) Null in Primary Key?

No, primary key cannot be null.

39) Null in Unique Key?

Yes, allowed.

40) Number of Nulls in Unique Key

Only one NULL allowed.

41) Constraints

Rules on table data. Types: Primary Key, Foreign Key, Unique, Not Null, Check, Default.

42) Composite Key

Combination of two or more columns as primary key.

43) View in SQL

Virtual table based on query result.

45) Foreign Key

Used to link two tables together (referential integrity).

46) Types of Join

1) Inner Join 2) Left Join 3) Right Join 4) Full Outer Join 5) Cross Join

47) Pattern Program

Use nested loops to print decreasing numbers in rows.

48) let vs var vs const (JS)

var → Function-scoped, can be redeclared let → Block-scoped, can be reassigned const → Block-scoped, cannot be reassigned

49) Hoisting

JavaScript default behavior of moving declarations to the top of scope.

50) Closure

Function with access to outer function variables even after outer function ends.

51) async and await (C#)

Used for asynchronous programming, await pauses execution until task completes.

52) Second Highest in Array

Sort array and pick second last OR track two largest values in one pass.

53) Series without Loop

Use recursion to generate and print values until step count is reached.