.NET and OOP Concepts

1. What is a Class?

A class is a blueprint for creating objects. It defines properties (variables) and methods (functions) that the object will have.

Example:

```
public class Car
{
    public string Color;
    public void Drive()
    {
        Console.WriteLine("The car is driving...");
    }
}
```

This class `Car` can be used to create car objects.

2. What is an Object?

An object is an instance of a class. It holds real-time values and can use methods of the class.

Example:

```
Car myCar = new Car();
myCar.Color = "Red";
myCar.Drive();  // Output: The car is driving...
```

3. Types of Authentication

- **Windows Authentication**: Uses Windows credentials.
- **Forms Authentication**: Login using username/password (e.g., Login pages).
- **JWT Authentication (Token-based)**: Common in APIs.
- **OAuth**: Login using Google, Facebook, etc.

Real-time Examples:

- Gmail uses OAuth.

- Your company intranet may use Windows Authentication.
- Your API may use JWT Bearer tokens.

4. OOP Concepts

- **Encapsulation**: Binding data with code.
- **Abstraction**: Hiding internal details.
- **Inheritance**: Reusing base class properties.
- **Polymorphism**: One function, many forms.

Real-time Example:

- You have a class `Vehicle`, and `Car`, `Bike` inherit from it.
- All `Drive()` methods behave differently in each child class → Polymorphism.

5. Versions of Currently Used Tools

- **.NET**: .NET 6 / 7 / 8 (latest)
- **Visual Studio**: VS 2022 / VS 2025
- **SQL Server**: 2019 / 2022
- **Angular**: v16 / v17
- **Entity Framework Core**: EF Core 6 / 7 / 8

Verification:

```
dotnet --version
```

6. What is Entity & Why We Use It?

An Entity is a class that represents a table in your database using Entity Framework.

Example:

```
public class Student
{
    public int Id { get; set; }
    public string Name { get; set; }
}
```

This maps to a table called `Students`. Entity is used to simplify DB operations without writing SQL.

7. Difference Between POST & GET

Feature	GET	POST
Purpose	Fetch data	Submit data
URL	Parameters visible in URL	Parameters in body
Idempotent	Yes	No
Secure	Less secure	More secure

Real-time Example:

- `GET /students/1` → fetch student with ID 1
- `POST /students` → insert new student record

8. Difference Between Entity & Stored Procedure

Feature	Entity Framework	Stored Procedure
Code Style	Object-oriented (C#)	SQL-based
Flexibility	Easier to maintain	Difficult to change
Performance	Slower for complex logic	Faster for complex logic
Use Case	CRUD operations	Complex joins/logic

Usage: Use Entity for normal CRUD, use Stored Procedure for optimized heavy logic.

9. .NET Core vs .NET Framework

Feature	.NET Framework	.NET Core (.NET 5/6/7/8)
OS Support	Windows only	Cross-platform (Win/Linux)
Performance	Slower	High performance
Development Focus	Desktop/Enterprise	Web, Microservices, Cloud

Recommendation: Always choose .NET Core / .NET 6+ for modern apps.

10. What is ADO.NET?

ADO.NET is a data access technology in .NET that uses SQL commands, connections, and readers to interact with the DB.

Example:

```
SqlConnection con = new SqlConnection("conn string");
SqlCommand cmd = new SqlCommand("SELECT * FROM Students", con);
SqlDataReader reader = cmd.ExecuteReader();
```

Use when you want fine control over DB logic.

11. What is Return Type in .NET?

Every method must define what it returns.

Examples:

```
public int Add(int a, int b)  // returns integer
{
    return a + b;
}

public void Print()  // returns nothing
{
    Console.WriteLine("Hello");
}
```

12. API Methods

Method	Usage
GET	Retrieve data
POST	Create data
PUT	Update entire resource
PATCH	Update partial resource
DELETE	Delete data

Example:

```
GET /api/products
POST /api/products
```

13. What is Razor Page?

Razor Pages is a simplified web UI framework in ASP.NET Core that uses `.cshtml` files (HTML + C# code together).

Example:

- `Index.cshtml` – UI
- `Index.cshtml.cs` – Code-behind

Use when building web apps with UI + backend together.

14. ASP.NET Life Cycle

For Web Forms:

- Page Request
- Start
- Initialization
- Load
- Postback

- Rendering
- Unload

In modern ASP.NET Core: Middleware pipeline → Controller → Action → Response

15. What is Namespace?

Namespace is a container for related classes.

Example:

```
namespace StudentApp.Models
{
    public class Student { }
}
```

Helps to organize code and avoid name conflicts.

16. What are Filters?

Filters in ASP.NET Core are used to run logic before or after controller actions.

Types:

- **Authorization Filter**: Check user roles
- **Action Filter**: Log or modify input/output
- **Exception Filter**: Handle errors

Example:

```
public class LogActionFilter : IActionFilter
{
    public void OnActionExecuting(...) { /* log */ }
    public void OnActionExecuted(...) { /* log */ }
}
```

17. What is LINQ?

LINQ (Language Integrated Query) lets you write SQL-like queries in C#.

Example:

```
var names = students.Where(s => s.Age > 18).Select(s => s.Name);
```

You can use LINQ on Arrays, Lists, DBs (via EF), XML, etc.