C# Topics

* Understanding the CLR (Common Language Runtime)
* Basic structure of a C# program (Main method, namespaces, assemblies
* Variables and Data Types (primitive types, reference types)
* Operators (arithmetic, relational, logical, etc.)
* Conditionals (if, else, switch)
* Loops (for, while, do-while, foreach)
* Methods and Functions
* Exception Handling (try-catch-finally, throw)
* Classes and Objects
* Encapsulation (properties, fields, access modifiers)
* Inheritance and Polymorphism (base classes, abstract classes, method overriding)
* Interfaces and implementation
* Constructors and destructors
* Static vs. Instance members
* Arrays and Lists
* Dictionaries, HashSets, Queues, and Stacks
* LINQ (Language Integrated Query)
* Working with Generics (generic methods, generic classes)
* IEnumerable, IEnumerator, and using the yield keyword
* Delegates and Events
* Lambda expressions
* Anonymous methods
* Properties (auto-implemented, computed properties)
* Extension methods
* Indexers and overloading operators
* Async and Await
* Task-based asynchronous pattern (TAP)
* Parallel programming (Tasks, Threads)
* Cancellation tokens and handling async exceptions
* Working with files (File, FileStream, StreamReader/StreamWriter)
* Reading and writing text/binary data
* Serialization (JSON, XML, BinaryFormatter)
* Using libraries like Newtonsoft.Json for JSON serialization
* The garbage collector (GC) and finalization
* Memory management patterns in C#
* IDisposable and the using statement
* Reference vs Value types and understanding boxing/unboxing
* Defining and using attributes
* Reflection (introspection of types, methods, and properties)
* Dynamic assembly loading
* Concept of Dependency Injection (DI) in C#

ASP.NET & .NET Core Topics

* Introduction to ASP.NET Core
* MVC (Model-View-Controller) pattern
* RESTful services with WebAPI
* Routing and middleware
* Razor Pages, Blazor, and working with Views
* Common design patterns (Singleton, Factory, Observer, Dependency Injection)
* SOLID principles
* Applying patterns in real-world scenarios
* Writing unit tests in C# using frameworks like MSTest, NUnit, or xUnit
* Mocking and dependency injection in tests
* TDD (Test-Driven Development)
* Calling external APIs using HttpClient
* Parsing JSON and XML data
* Creating and securing APIs (JWT, OAuth)
* ADO.NET basics (SQL commands, DataReader, DataAdapter)
* ORM (Object-Relational Mapping) with Entity Framework
* LINQ-to-SQL and LINQ-to-Entities
* Asynchronous database operations
* Using built-in DI containers in .NET Core/ASP.NET Core
* Understanding IoC (Inversion of Control) and services