

DOT NET FULL INTERVIEW NOTES (BEGINNER TO ADVANCED)

1. What is .NET?

.NET is a Microsoft framework used to build Web, Desktop, API, Cloud, IoT, and Gaming applications. It supports C#, VB.NET, F#, and runs cross-platform using .NET Core and .NET 5+ unified platform.

.NET Framework vs .NET Core vs .NET (Unified)

- .NET Framework: Windows-only, older. - .NET Core: Fast, secure, cross-platform. - .NET 5/6/7/8+: Unified, modern, high-performance.

CLR (Common Language Runtime)

CLR runs and manages .NET applications. Handles memory management, garbage collection, exception handling, and JIT compilation.

2. C# COMPLETE BASICS

Data Types in C#

Value Types: int, float, double, char, bool, struct, enum. Reference Types: string, array, class, interface, object.

Keywords

static, const, readonly, sealed, var, dynamic, async, await.

Control Statements

if, else, switch, for, foreach, while, do-while, break, continue.

3. OOPS in C# (MOST IMPORTANT TOPIC)

Encapsulation

Wrapping data + methods. Implemented using private fields + public properties.

Abstraction

Hide implementation details using abstract classes and interfaces.

Inheritance

One class inherits from another using ':' symbol. C# supports single inheritance but multiple interface inheritance.

Polymorphism

Compile-time (method overloading) and runtime (method overriding using virtual/override).

4. IMPORTANT ADVANCED C# CONCEPTS

Constructors

Types: default, parameterized, copy, static, private constructors.

Collections in C#

List, Dictionary, HashSet, Queue, Stack. Generic collections are type-safe.

ASYNC & AWAIT

Used for asynchronous programming. Improves performance & responsiveness.

5. ASP.NET CORE (WEB DEVELOPMENT FRAMEWORK)

What is ASP.NET Core?

Modern, open-source, cross-platform framework to build web apps and APIs.

MVC Architecture

Model: Database + business logic View: UI Controller: Request handling + action methods.

Middleware Pipeline

Every ASP.NET Core application uses a sequence of middleware. Example: Authentication, Routing, CORS.

Dependency Injection (DI)

Built-in DI. Reduces tight coupling and improves testing.

6. ENTITY FRAMEWORK CORE (DATABASE ORM)

What is EF Core?

EF Core is ORM to interact with database using C# code instead of SQL.

Migration Commands

add-migration update-database remove-migration

LINQ Important Queries

Where, Select, OrderBy, GroupBy, Join, FirstOrDefault, Count, Any.

7. REST API DEVELOPMENT

HTTP Methods

GET, POST, PUT, PATCH, DELETE.

Status Codes

200 OK, 201 Created, 400 Bad Request, 401 Unauthorized, 404 Not Found, 500 Server Error.

JWT Authentication

Used for secure login. Consists of Header + Payload + Signature.

8. SQL BASICS (IMPORTANT FOR .NET INTERVIEW)

SELECT, INSERT, UPDATE, DELETE, JOINS (INNER, LEFT, RIGHT, FULL), GROUP BY, HAVING, PRIMARY KEY, FOREIGN KEY, Normalization.

9. TOP .NET INTERVIEW QUESTIONS

1. What is .NET CLR? 2. Difference between .NET Core and .NET Framework? 3. What is Dependency Injection? 4. What is Middleware? 5. What is the difference between Interface and Abstract class? 6. What is Polymorphism? 7. What is EF Core Migration? 8. Explain REST vs SOAP. 9. Explain async and await. 10. How JWT works?

10. 1-MONTH STUDY PLAN

Week 1: C# basics + OOPs Week 2: Collections, Delegates, Async, MVC basics Week 3: SQL + EF Core + CRUD Week 4: REST API + Authentication + Interview Q&A;