**1. What is a Memory Leak in .NET and how can you detect it?**

**Answer: A memory leak occurs when objects are not properly disposed or dereferenced, preventing the garbage collector from reclaiming memory. Tools like Visual Studio Diagnostic Tools or dotMemory help detect such leaks.**

**2. What is the difference between Task.Run and async/await?**

**Answer: Task.Run is used to offload CPU-bound work to a separate thread, while async/await is ideal for non-blocking I/O operations. Overusing Task.Run can result in unnecessary thread creation.**

**3. How do you implement Role-based Authorization in ASP.NET Core?**

**Answer: Use the [Authorize(Roles = "Admin")] attribute or configure policies in Startup.cs with AddAuthorization. Assign roles via claims or Identity framework.**

**4. What are the key differences between .NET Core and .NET Framework?**

**Answer:**

* **.NET Core is cross-platform, modular, and open-source**
* **.NET Framework is Windows-only and monolithic**
* **.NET Core supports CLI tools and is more performance-oriented**

**5. What is the difference between transient, scoped, and singleton lifetimes?**

**Answer:**

* **Transient: A new instance every time it's requested**
* **Scoped: One instance per request scope**
* **Singleton: A single instance throughout the application lifetime**

**6. How do you implement Soft Delete in EF Core?**

**Answer: Add a boolean IsDeleted field in the model. Override SaveChanges() or use global query filters to exclude soft-deleted records in queries.**

**7. What are the benefits of using Swagger in Web API?**

**Answer:**

* **Auto-generates API documentation**
* **Provides test UI for endpoints**
* **Helps consumers understand request/response schema**
* **Useful for client code generation**

**8. What is CORS and how do you enable it in ASP.NET Core?**

**Answer: CORS (Cross-Origin Resource Sharing) allows a web app from one domain to access resources from another. It is configured using services.AddCors() and app.UseCors() in Startup.cs.**

**9. What are Action Results in ASP.NET Core MVC?**

**Answer: Action results are return types from controller actions that represent HTTP responses, e.g., Ok(), BadRequest(), Content(), Json(), File().**

**10. How do you implement Rate Limiting in a Web API?**

**Answer: Use middleware or tools like AspNetCoreRateLimit NuGet package. Configure rate limit policies based on IP, endpoint, or user identity.**

**11. How do you perform unit testing with EF Core?**

**Answer: Use InMemoryDatabase provider for EF Core, mock DbContext or use repository abstraction, and test against expected state or behavior.**

**12. Explain Managed Identity in Azure.**

**Answer: Managed Identity allows Azure resources to authenticate to services (like Key Vault, Storage) without managing credentials. Enabled through Azure Portal or ARM templates.**

**13. What are Durable Functions in Azure?**

**Answer: Durable Functions are stateful workflows built on Azure Functions using Orchestrator, Activity, and Client functions. Ideal for long-running operations and chaining.**

**14. How do you optimize LINQ queries for performance?**

**Answer:**

* **Filter early using Where**
* **Use projection (Select) to reduce data transfer**
* **Avoid in-memory operations by using IQueryable**
* **Use indexes in DB and avoid ToList() until necessary**

**15. Code Challenge: Remove duplicates from a list of integers**

**Answer:**

**public List<int> RemoveDuplicates(List<int> numbers)**

**{**

**return numbers.Distinct().ToList();**

**}**