.NET Important Topics with Concept Explanation & 40 MCQ Notes

---

📄 Detailed Concept Explanation (Key Topics)

1. System.Collections.Generic Namespace

- Contains generic collection classes like List<T>, Dictionary<TKey, TValue>, Stack<T>, Queue<T>, etc.
- Generic collections offer type-safety and performance benefits.

2. Delegates in C#

- A delegate is a type-safe function pointer used for callbacks and event handling.
- Used to implement runtime notification/event systems.
- Syntax: delegate void MyDelegate(string msg);

3. Value Type vs Reference Type

- **Value Types**: store actual data (e.g., int, float, bool). Stored on stack.
- **Reference Types**: store reference to data (e.g., string, object, class). Stored on heap.

4. Interfaces

- Define contracts (method/property signatures) without implementation.
- A class implementing an interface must provide the method bodies.
- Supports multiple inheritance indirectly.

5. Exception Handling

- try-catch-finally used for handling runtime errors.
- finally always executes (for cleanups).

6. OOP Concepts in C#

- Inheritance, Polymorphism (Compile-time & Runtime), Encapsulation, Abstraction
- Method Overloading (same method name, different params)

7. Static, Sealed, Abstract Keywords

- sealed: Prevents further inheritance
- abstract: Can't instantiate; contains abstract methods

- static: Belongs to class, not instance

8. Multithreading & Thread Pool

- Thread pool reuses threads, improves performance
- Task, Thread, async/await used for parallelism

9. CLR Security

- Common Language Runtime provides Code Access Security (CAS), Role-Based Security

10. FileStream

- Used for reading/writing binary/text files in .NET
- Located under System.IO

11. ASP.NET Core Concepts

- Kestrel server, Razor engine, MVC pattern (Model-View-Controller)
- Middleware: UseStaticFiles(), UseRouting()
- Lifetimes: Singleton, Scoped, Transient

12. Entity Framework (EF)

- ORM for database operations
- DbContext, DbSet<T> for querying and saving data

13. Model Binding & Data Annotations

- Automatically maps request data to action parameters
- Validation using [Required], [StringLength], etc.

🔹 **40 Important MCQs with Answers & Explanation**

1. **Identify the class belonging to System.Collections.Generic:**
   ✅ **Answer: Stack**
   🧠 Stack<T> is part of System.Collections.Generic.
2. **Which construct is suitable for notification system?**
   ✅ **Answer: Delegate**
   🧠 Delegates are used for callback/event notification.
3. **Condition for implementing delegates?**
   ✅ **Answer: Inheritance**
   🧠 Delegates work in polymorphic patterns, relying on inheritance.
4. **Difference between value and reference type?**
   ✅ **Answer: Value type stores value, reference type stores pointer.**
5. **Purpose of interface in C#?**
   ✅ **Answer: Define a contract of methods/properties**
6. **Output of delegate with ref string:**
   ✅ **Answer: ur C#.NET Skills**
7. **Output of method overload in inheritance?**
   ✅ **Answer: Abc from A** (int matches base class method)
8. **Custom exception declaration:**
   ✅ **Answer: class InvalidResponse : Exception {}
9. **Modifier to prevent inheritance:**
   ✅ **Answer: Sealed**
10. **Output of Array.Reverse:**
    ✅ **Answer: World** (array reversed)
11. **Concurrency mechanism without new thread:**
    ✅ **Answer: thread pool**
12. **CLR security contribution:**
    ✅ **Answer: Code access and role-based security**
13. **Tool to view IL code:**
    ✅ **Answer: ildasm.exe**
14. **Purpose of async keyword:**
    ✅ **Answer: Enables use of await keyword**

15. **Managed vs unmanaged code:**
    ✅ **Answer: i-True, ii-False, iii-True**

16. **When to use static local functions?**
    ✅ **Answer: When used only inside specific method**

17. **Output of Dog inheriting Animal:**
    ✅ **Answer: Dog is barking. Animal is eating.**

18. **True about custom property:**
    ✅ **Answer: Custom setter doubles if value > 0**

19. **FileStream in .NET:**
    ✅ **Answer: Used for both reading and writing**

20. **Purpose of custom attributes:**
    ✅ **Answer: Add metadata to types/members**

21. **ASP.NET Core hosting options:**
    ✅ **Answer: All of the Above**

22. **Files for localization:**
    ✅ **Answer: .resx**

23. **Lifetime for stateless/lightweight services:**
    ✅ **Answer: Transient**

24. **Service method supporting full MVC:**
    ✅ **Answer: AddControllersWithViews()**

25. **Middleware to serve static files:**
    ✅ **Answer: app.UseStaticFiles()**

26. **Memory-resident ADO.NET object:**
    ✅ **Answer: DataSet**

27. **EF class for entity set:**
    ✅ **Answer: DbSet**

28. **EF Core development approach:**
    ✅ **Answer: Both A and B (Code-first, DB-first)**

29. **Base class for API Controller:**
    ✅ **Answer: ControllerBase**

30. **Every layout must call:**
    ✅ **Answer: @RenderBody**

31. **Business logic should go in:**
    ✅ **Answer: Model**

32. **Who creates ViewModel?**
    ✅ **Answer: Controller, Model**

33. **Common controller logic goes in:**
    ✅ **Answer: Filters**

34. **Route definitions are placed next to:**
    ✅ **Answer: Controller, Action**

35. **Converting request data to objects:**
    ✅ **Answer: Model Binding**

36. **ASP.NET Core validation uses:**
    ✅ **Answer: Model Object**

37. **Validation attributes checked on:**
    ✅ **Answer: before controller action**

38. **Encapsulate crosscutting concerns:**
    ✅ **Answer: Filters**

39. **Server-side dynamic content using:**
    ✅ **Answer: Razor View**

40. **Razor engine allows defining:**
    ✅ **Answer: layouts, partial views**