

# SESSION 1 – .NET FRAMEWORK (CCEE MCQs)

---

## EASY LEVEL (20 MCQs)

### 1. What is the main purpose of the .NET Framework?

- A) To compile Java programs
- B) To provide a platform for building Windows and web applications
- C) To replace operating systems
- D) To manage databases

 Answer: B

---

### 2. Which language is NOT supported by .NET?

- A) C#
- B) VB.NET
- C) F#
- D) C

 Answer: D

---

### 3. What does CLR stand for?

- A) Common Language Runtime
- B) Common Logic Runtime
- C) Core Language Runtime
- D) Code Language Runtime

 Answer: A

---

### 4. What is IL in .NET?

- A) High-level language
- B) Machine code
- C) Intermediate language generated after compilation
- D) Assembly language

 Answer: C

---

**5. Which component executes IL code?**

- A) Compiler
- B) CLR
- C) OS
- D) Loader

 Answer: B

---

**6. Which file extension is used for executable assemblies?**

- A) .dll
- B) .exe
- C) .cs
- D) .il

 Answer: B

---

**7. What is an assembly?**

- A) A namespace
- B) A compiled unit of code
- C) A class
- D) A method

 Answer: B

---

**8. Which component performs Garbage Collection?**

- A) JIT
- B) CTS
- C) CLR
- D) CLS

 Answer: C

---

**9. What does CTS define?**

- A) Language syntax**
- B) Data types supported by .NET**
- C) Assembly metadata**
- D) Security rules**

 **Answer:** B

---

## 10. What does CLS ensure?

- A) Security**
- B) Memory management**
- C) Language interoperability**
- D) Performance**

 **Answer:** C

---

## 11. Which unit contains metadata?

- A) Namespace**
- B) Assembly**
- C) Method**
- D) Class**

 **Answer:** B

---

## 12. What is JIT compilation?

- A) Compile at install time**
- B) Compile at runtime**
- C) Compile before execution only**
- D) Manual compilation**

 **Answer:** B

---

## 13. Which component converts IL to native code?

- A) CLR**
- B) CTS**
- C) CLS**
- D) JIT**

 **Answer:** D

---

## **14. What type of code does CLR execute?**

- A) Source code**
- B) IL code**
- C) Bytecode**
- D) Assembly code**

 **Answer:** B

---

## **15. Which file contains reusable code?**

- A) .exe**
- B) .dll**
- C) .txt**
- D) .config**

 **Answer:** B

---

## **16. AppDomain is mainly used for?**

- A) UI separation**
- B) Memory isolation**
- C) Database access**
- D) Thread creation**

 **Answer:** B

---

## **17. Which feature prevents memory leaks?**

- A) CTS**
- B) CLS**
- C) Garbage Collection**
- D) JIT**

 **Answer:** C

---

## **18. Which component manages security?**

- A) CLR**
- B) JIT**

- C) CTS**
- D) Compiler**

 **Answer:** A

---

### **19. Which code is platform-independent?**

- A) Native code**
- B) IL code**
- C) Machine code**
- D) Assembly code**

 **Answer:** B

---

### **20. Which language rule ensures cross-language compatibility?**

- A) CTS**
- B) CLS**
- C) IL**
- D) JIT**

 **Answer:** B

---

## **MEDIUM LEVEL (20 MCQs)**

### **21. What happens first when a .NET program runs?**

- A) JIT compilation**
- B) CLR loads the assembly**
- C) Garbage Collection**
- D) CTS validation**

 **Answer:** B

---

### **22. Which assembly contains entry point?**

- A) DLL**
- B) EXE**
- C) Shared assembly**
- D) Satellite assembly**

 **Answer:** B

---

### **23. IL code is generated by?**

- A) CLR**
- B) JIT**
- C) Language compiler**
- D) OS**

 **Answer:** C

---

### **24. CTS supports which data type category?**

- A) Only reference types**
- B) Only value types**
- C) Both value and reference types**
- D) Primitive types only**

 **Answer:** C

---

### **25. CLS is a subset of?**

- A) IL**
- B) CLR**
- C) CTS**
- D) JIT**

 **Answer:** C

---

### **26. Which metadata is stored in assembly?**

- A) Variable values**
- B) Type definitions**
- C) Runtime stack**
- D) Heap objects**

 **Answer:** B

---

### **27. What triggers Garbage Collection?**

- A) Manual call only**
- B) Low memory condition**

- C) Program termination
- D) Thread stop

 Answer: B

---

## 28. Which memory area stores objects?

- A) Stack
- B) Heap
- C) Register
- D) Cache

 Answer: B

---

## 29. AppDomains help in?

- A) Security isolation
- B) Faster execution
- C) Code compilation
- D) Thread pooling

 Answer: A

---

## 30. Which code executes faster?

- A) IL code
- B) Native code
- C) Source code
- D) Bytecode

 Answer: B

---

## 31. What happens after JIT compilation?

- A) IL is deleted
- B) Native code is cached
- C) Program terminates
- D) CLR stops

 Answer: B

---

### **32. Which assembly can be shared across applications?**

- A) Private assembly**
- B) Static assembly**
- C) Shared assembly**
- D) Dynamic assembly**

 **Answer:** C

---

### **33. Which component enforces type safety?**

- A) CLR**
- B) CTS**
- C) CLS**
- D) JIT**

 **Answer:** A

---

### **34. What ensures safe execution of code?**

- A) JIT**
- B) CLR**
- C) Compiler**
- D) OS**

 **Answer:** B

---

### **35. Which feature allows multiple languages to work together?**

- A) IL**
- B) CLS**
- C) JIT**
- D) AppDomain**

 **Answer:** B

---

### **36. Garbage Collector works on which principle?**

- A) Reference counting**
- B) Mark and sweep**
- C) Manual delete**
- D) FIFO**

 Answer: B

---

### 37. Which part manages threads?

- A) CTS
- B) CLS
- C) CLR
- D) IL

 Answer: C

---

### 38. What is stored in metadata?

- A) Business logic
- B) Assembly info
- C) Runtime variables
- D) Heap references

 Answer: B

---

### 39. Which process isolates applications?

- A) Thread
- B) AppDomain
- C) Assembly
- D) Namespace

 Answer: B

---

### 40. Which compilation happens per method?

- A) Static compilation
- B) AOT
- C) JIT compilation
- D) Pre-compilation

 Answer: C

---

 HARD LEVEL (20 MCQs – TRICKY)

#### **41. If CLS rules are violated, what happens?**

- A) Program won't compile**
- B) Runtime error**
- C) Other languages can't use it**
- D) CLR crashes**

 **Answer:** C

---

#### **42. Which assembly cannot run independently?**

- A) EXE**
- B) DLL**
- C) Shared assembly**
- D) Strong named assembly**

 **Answer:** B

---

#### **43. Which statement about IL is TRUE?**

- A) Platform dependent**
- B) CPU specific**
- C) Platform independent**
- D) OS dependent**

 **Answer:** C

---

#### **44. Garbage Collection does NOT manage?**

- A) Managed heap**
- B) Unmanaged memory**
- C) Object lifetime**
- D) Memory compaction**

 **Answer:** B

---

#### **45. Which is NOT a CLR responsibility?**

- A) Security**
- B) Exception handling**
- C) UI rendering**
- D) Memory management**

 Answer: C

---

#### 46. What happens if AppDomain crashes?

- A) Whole process crashes
- B) Only that AppDomain unloads
- C) CLR stops
- D) OS shuts down

 Answer: B

---

#### 47. JIT compiles IL into?

- A) Bytecode
- B) Source code
- C) Native machine code
- D) Assembly

 Answer: C

---

#### 48. Which improves performance after first execution?

- A) CLR
- B) Cached native code
- C) IL regeneration
- D) GC

 Answer: B

---

#### 49. CTS allows which feature?

- A) Language-specific syntax
- B) Cross-language type usage
- C) Faster execution
- D) Better UI

 Answer: B

---

#### 50. Which causes delayed GC execution?

- A) Immediate memory request**
- B) High available memory**
- C) Manual call**
- D) Thread creation**

 **Answer:** B

---

## **51. Which code is verified by CLR?**

- A) Native code**
- B) Managed code**
- C) Assembly code**
- D) OS code**

 **Answer:** B

---

## **52. Which type of assembly contains entry point?**

- A) Class library**
- B) DLL**
- C) EXE**
- D) Satellite**

 **Answer:** C

---

## **53. Which is NOT part of assembly structure?**

- A) Manifest**
- B) Metadata**
- C) IL code**
- D) Source code**

 **Answer:** D

---

## **54. Which feature enforces security boundaries?**

- A) CTS**
- B) CLS**
- C) AppDomain**
- D) IL**

 **Answer:** C

---

**55. Which GC generation is collected most frequently?**

- A) Gen 0
- B) Gen 1
- C) Gen 2
- D) LOH

 Answer: A

---

**56. Which ensures language neutrality?**

- A) CLR
- B) IL
- C) CTS
- D) CLS

 Answer: C

---

**57. Which improves memory efficiency?**

- A) Stack allocation
- B) Heap compaction
- C) IL generation
- D) Thread pooling

 Answer: B

---

**58. Which statement is FALSE?**

- A) CLR manages threads
- B) IL is CPU independent
- C) JIT runs at compile time
- D) GC is automatic

 Answer: C

---

**59. Which exception is handled by CLR?**

- A) Hardware interrupt
- B) Runtime exception

- C) Compile-time error
- D) Syntax error

 Answer: B

---

## 60. Which ensures safe execution across languages?

- A) JIT
- B) CTS + CLS
- C) GC
- D) IL

 Answer: B

# SESSION 2 – .NET PLATFORM & TOOLS (CCEE MCQs)

---

## EASY LEVEL (10 MCQs)

### 1. Which framework is cross-platform by default?

- A) .NET Framework
- B) .NET Core
- C) Mono
- D) Xamarin

 Answer: B

---

### 2. .NET Framework mainly runs on which OS?

- A) Linux
- B) macOS
- C) Windows
- D) Android

 Answer: C

---

### **3. Which framework is mainly used for mobile application development?**

- A) .NET Framework
- B) .NET Core
- C) Xamarin
- D) ASP.NET

 **Answer:** C

---

### **4. What type of code runs under CLR control?**

- A) Unmanaged code
- B) Native code
- C) Managed code
- D) Machine code

 **Answer:** C

---

### **5. Which IDE is officially used for .NET development?**

- A) Eclipse
- B) IntelliJ
- C) Visual Studio
- D) NetBeans

 **Answer:** C

---

### **6. Which tool is used to view IL code?**

- A) JIT
- B) CLR
- C) ILDASM
- D) CTS

 **Answer:** C

---

### **7. Mono is mainly designed for?**

- A) Windows only
- B) Cross-platform .NET support
- C) Database management
- D) Game engines

 **Answer:** B

---

## **8. Which code is directly executed by OS?**

- A) IL code
- B) Managed code
- C) Unmanaged code
- D) C# code

 **Answer:** C

---

## **9. Which Visual Studio feature helps in debugging?**

- A) ILDASM
- B) Breakpoints
- C) CLR
- D) CTS

 **Answer:** B

---

## **10. .NET Core is optimized for?**

- A) Desktop apps only
- B) Web and cloud apps
- C) Mobile apps only
- D) Embedded systems

 **Answer:** B

---

## **MEDIUM LEVEL (10 MCQs)**

### **11. Which framework replaced .NET Core?**

- A) Mono
- B) Xamarin
- C) .NET 5+
- D) .NET Framework

 **Answer:** C

---

## **12. Which is TRUE about managed code?**

- A) Runs without CLR
- B) Requires manual memory management
- C) Runs under CLR supervision
- D) Cannot use garbage collection

 **Answer:** C

---

## **13. Which framework supports side-by-side versioning?**

- A) .NET Framework
- B) .NET Core
- C) Mono
- D) Xamarin

 **Answer:** B

---

## **14. What does ILDASM stand for?**

- A) Intermediate Language Debugger
- B) Intermediate Language Disassembler
- C) Integrated Language Disassembler
- D) Internal Language Debugger

 **Answer:** B

---

## **15. Which framework is best suited for microservices?**

- A) .NET Framework
- B) Xamarin
- C) .NET Core
- D) WinForms

 **Answer:** C

---

## **16. Which code requires explicit memory deallocation?**

- A) Managed code
- B) IL code
- C) Unmanaged code
- D) C# code

 Answer: C

---

### 17. Visual Studio internally uses which compiler for C#?

- A) JIT
- B) Roslyn
- C) CLR
- D) CTS

 Answer: B

---

### 18. Which framework allows running .NET apps on Linux?

- A) .NET Framework
- B) Mono
- C) WinForms
- D) WPF

 Answer: B

---

### 19. What does ILDASM mainly display?

- A) Source code
- B) Native code
- C) IL + metadata
- D) Machine instructions

 Answer: C

---

### 20. Which framework is tightly coupled with Windows APIs?

- A) .NET Core
- B) Mono
- C) Xamarin
- D) .NET Framework

 Answer: D

---

 HARD LEVEL (10 MCQs – TRICKY)

## **21. Which statement is FALSE?**

- A) Managed code uses garbage collection
- B) Unmanaged code runs without CLR
- C) ILDASM modifies IL code
- D) .NET Core is cross-platform

 **Answer:** C

---

## **22. Why unmanaged code executes faster initially?**

- A) Uses CLR
- B) No runtime checks
- C) Uses IL
- D) Has metadata

 **Answer:** B

---

## **23. Which framework was merged into modern .NET?**

- A) Mono
- B) Xamarin
- C) .NET Core
- D) Silverlight

 **Answer:** C

---

## **24. Which versioning issue existed in .NET Framework?**

- A) Side-by-side execution
- B) DLL Hell
- C) Cross-platform issues
- D) Slow JIT

 **Answer:** B

---

## **25. Which application type CANNOT be built using .NET Core?**

- A) Console app
- B) Web API
- C) Windows Forms (initially)
- D) Cloud services

 Answer: C

---

## 26. Which ILDASM feature helps analyze security?

- A) Stack trace
- B) Manifest view
- C) Native code
- D) Debug symbols

 Answer: B

---

## 27. Which runs outside CLR boundary?

- A) Managed heap
- B) Garbage Collector
- C) Unmanaged DLL
- D) CTS

 Answer: C

---

## 28. Which tool converts IL back to readable form?

- A) JIT
- B) CLR
- C) ILDASM
- D) Roslyn

 Answer: C

---

## 29. Which framework is BEST for cloud-native apps?

- A) .NET Framework
- B) Xamarin
- C) .NET Core
- D) Mono

 Answer: C

---

## 30. Which combination causes memory leaks?

- A) Managed code + GC
- B) Unmanaged resources not released
- C) JIT compilation
- D) IL execution

 Answer: B

## SESSION 3 – C# BASICS & OOP (CCEE MCQs)

---

### EASY LEVEL (30 MCQs)

#### 1. Which project type is used to create reusable code?

- A) Console Application
- B) Class Library
- C) Web API
- D) MVC

 Answer: B

---

#### 2. Which keyword is used to include namespaces?

- A) include
- B) using
- C) import
- D) require

 Answer: B

---

#### 3. Which data type maps to `System.Int32`?

- A) int
- B) long
- C) short
- D) byte

 Answer: A

---

**4. Which method executes first in a console app?**

- A) Constructor
- B) Main
- C) Destructor
- D) Static block

 Answer: B

---

**5. Which keyword is used to create an object?**

- A) class
- B) new
- C) create
- D) this

 Answer: B

---

**6. Which CTS type represents text?**

- A) char
- B) string
- C) System.String
- D) Both B and C

 Answer: D

---

**7. Which keyword allows multiple parameters?**

- A) args
- B) ref
- C) params
- D) out

 Answer: C

---

**8. Which allows calling method with parameter names?**

- A) Optional parameters
- B) Named parameters
- C) Positional parameters
- D) Params

 **Answer:** B

---

## **9. Which property accessor retrieves value?**

- A) set
- B) get
- C) init
- D) readonly

 **Answer:** B

---

## **10. Which keyword creates a read-only property?**

- A) static
- B) readonly
- C) const
- D) private

 **Answer:** B

---

## **11. Constructors are used to?**

- A) Destroy objects
- B) Initialize objects
- C) Free memory
- D) Call methods

 **Answer:** B

---

## **12. Destructor syntax starts with?**

- A) delete
- B) ~
- C) !
- D) dispose

 **Answer:** B

---

### **13. Which method cleans unmanaged resources?**

- A) Finalize
- B) Dispose
- C) Clear
- D) Free

 **Answer:** B

---

### **14. Which feature allows same method name?**

- A) Overriding
- B) Overloading
- C) Shadowing
- D) Hiding

 **Answer:** B

---

### **15. Which keyword allows passing variable arguments?**

- A) ref
- B) out
- C) params
- D) in

 **Answer:** C

---

### **16. Object Initializer removes need of?**

- A) Properties
- B) Fields
- C) Constructor parameters
- D) Methods

 **Answer:** C

---

### **17. Which keyword allows local methods?**

- A) private
- B) static

- C) void
- D) function

 **Answer:** B

---

### **18. Which property has only get?**

- A) Writable
- B) Read-only
- C) Static
- D) Virtual

 **Answer:** B

---

### **19. Which maps to `System.Boolean`?**

- A) bit
- B) bool
- C) Boolean
- D) logical

 **Answer:** B

---

### **20. Which statement is true?**

- A) Destructor is deterministic
- B) Dispose is deterministic
- C) GC calls Dispose
- D) Destructor frees managed memory

 **Answer:** B

---

### **21. Which is required for `IDisposable`?**

- A) Destructor
- B) Dispose method
- C) GC.Collect
- D) Finalize

 **Answer:** B

---

## **22. Which keyword initializes object properties?**

- A) init
- B) this
- C) new
- D) set

 **Answer:** C

---

## **23. Which is CTS-compliant?**

- A) Pointer types
- B) unsigned int
- C) CLS rules
- D) System types

 **Answer:** D

---

## **24. Which type is value type?**

- A) string
- B) object
- C) struct
- D) class

 **Answer:** C

---

## **25. Which property cannot be modified?**

- A) Read-only
- B) Auto
- C) Public
- D) Static

 **Answer:** A

---

## **26. Which parameter is optional?**

- A) Without default value
- B) With default value
- C) ref parameter
- D) params

 **Answer:** B

---

**27. Which method can be inside another method?**

- A) Static
- B) Local function
- C) Constructor
- D) Destructor

 **Answer:** B

---

**28. Which is NOT a property accessor?**

- A) get
- B) set
- C) init
- D) fetch

 **Answer:** D

---

**29. Which keyword releases resources manually?**

- A) finalize
- B) dispose
- C) delete
- D) clear

 **Answer:** B

---

**30. Which project outputs DLL?**

- A) Console App
- B) Class Library
- C) Web App
- D) Razor App

 **Answer:** B

---

# MEDIUM LEVEL (30 MCQs – MANY SNIPPETS)

31.

```
void Test(int x = 10) {  
    Console.WriteLine(x);  
}  
Test();
```

Output?

- A) 0
- B) 10
- C) Error
- D) null

 Answer: B

---

32.

```
void Sum(params int[] x) {  
    Console.WriteLine(x.Length);  
}  
Sum(1,2,3);
```

Output?

- A) 2
- B) 3
- C) Error
- D) 0

 Answer: B

---

33. Which call is valid?

```
void M(int a, int b=5)
```

- A) M()
- B) M(1)
- C) M(b:2)
- D) M()

**Answer:** B

---

**34.**

```
class A {  
    public int X { get; }  
    A(int x){ X=x; }  
}
```

Which is true?

- A) Writable
- B) Compile error
- C) Read-only property
- D) Needs setter

**Answer:** C

---

**35.**

```
var obj = new Student { Id=1, Name="A" };
```

What is used?

- A) Constructor
- B) Object initializer
- C) Reflection
- D) Static init

**Answer:** B

---

**36.**

```
void M(int a, int b)  
void M(int a)
```

This is?

- A) Override
- B) Overload
- C) Hide
- D) Error

**Answer:** B

---

**37.**

```
using System;
```

Purpose?

- A) Memory allocation
- B) Namespace inclusion
- C) Assembly load
- D) GC trigger

 **Answer:** B

---

**38.**

```
void A(int x)
void A(ref int x)
```

Is this allowed?

- A) Yes
- B) No
- C) Runtime error
- D) CLS violation

 **Answer:** A

---

**39.**

```
~Test(){}  
~~~~~
```

Which method?

- A) Constructor
- B) Destructor
- C) Dispose
- D) Finalizer

 **Answer:** D

---

**40.**

```
class A : IDisposable { }
```

Missing?

- A) Destructor
- B) Dispose method
- C) Constructor
- D) Property

 Answer: B

---

#### 41. Which executes deterministically?

- A) Destructor
- B) GC
- C) Dispose
- D) Finalizer

 Answer: C

---

#### 42.

```
int x = default;
```

Value?

- A) null
- B) 1
- C) 0
- D) Error

 Answer: C

---

#### 43.

```
void M(int x, int y)  
M(y:2, x:1);
```

Result?

- A) Error
- B) 1 2
- C) 2 1
- D) Depends

 Answer: B

---

#### **44. Which allows flexible argument count?**

- A) ref
- B) out
- C) params
- D) in

 **Answer:** C

---

#### **45.**

```
int a;  
Console.WriteLine(a);
```

Result?

- A) 0
- B) null
- C) Compile error
- D) Runtime error

 **Answer:** C

---

#### **46. Which CTS type maps to string?**

- A) System.Object
- B) System.Text
- C) System.String
- D) System.Char

 **Answer:** C

---

#### **47.**

```
int Add(int a,int b=5)  
Add(10);
```

Result?

- A) 10
- B) 15
- C) Error
- D) 5

 **Answer:** B

---

## 48. Which cannot be overloaded?

- A) Constructors
- B) Methods
- C) Destructors
- D) Operators

 Answer: C

---

## 49. Which improves readability?

- A) Positional params
- B) Named params
- C) Params
- D) Ref

 Answer: B

---

## 50. Which ensures cleanup if Dispose not called?

- A) GC
- B) Finalizer
- C) Constructor
- D) CTS

 Answer: B

---

## 51. Which supports encapsulation?

- A) Fields
- B) Properties
- C) Methods
- D) Namespaces

 Answer: B

---

## 52. Which is CLS-compliant?

- A) Unsigned types
- B) Pointer types

- C) Method overloading
- D) Case-sensitive names

 Answer: C

---

### 53. Local functions improve?

- A) Memory
- B) Performance
- C) Encapsulation
- D) Security

 Answer: C

---

### 54. Which is true?

- A) Destructor calls Dispose
- B) Dispose calls GC
- C) Finalizer is non-deterministic
- D) GC frees unmanaged memory

 Answer: C

---

### 55. Which is a reference type?

- A) int
- B) bool
- C) struct
- D) class

 Answer: D

---

### 56. Which creates immutable state?

- A) set
- B) readonly property
- C) public field
- D) ref param

 Answer: B

---

## **57. Which can be overloaded?**

- A) Property
- B) Constructor
- C) Destructor
- D) Field

 **Answer:** B

---

## **58. Which avoids constructor overload explosion?**

- A) params
- B) Object initializer
- C) ref
- D) out

 **Answer:** B

---

## **59. Which is called automatically by GC?**

- A) Dispose
- B) Destructor
- C) Constructor
- D) Init

 **Answer:** B

---

## **60. Which statement is FALSE?**

- A) Dispose is manual
- B) Destructor is automatic
- C) GC is deterministic
- D) Finalizer is slow

 **Answer:** C

---

# **HARD LEVEL (30 MCQs – TRICKY & SNIPPETS)**

**61.**

```
void M(int a, params int[] b)
M(1);
```

Value of b.Length?

- A) 0
- B) 1
- C) null
- D) Error

 **Answer:** A

---

**62.**

```
class A {
    public int X { get; private set; }
}
```

Which is true?

- A) Fully read-only
- B) Writable inside class
- C) Compile error
- D) Immutable

 **Answer:** B

---

**63.**

```
void M(int x=1,int y)
```

Is this valid?

- A) Yes
- B) No
- C) Runtime error
- D) CLS violation

 **Answer:** B

---

**64.**

```
Student s = new();
```

Requires?

- A) Parameterized constructor
- B) Default constructor
- C) Object initializer
- D) Dispose

 **Answer:** B

---

**65.**

```
using(var f = new FileStream())
```

What happens?

- A) GC runs
- B) Dispose auto-called
- C) Finalizer runs
- D) Memory leak

 **Answer:** B

---

**66. Which breaks CLS?**

- A) Method overloading
- B) Unsigned int
- C) Named params
- D) Object initializer

 **Answer:** B

---

**67. Which is NOT true?**

- A) Destructor maps to Finalize
- B) Dispose is deterministic
- C) GC calls Dispose
- D) Finalizer is slow

 **Answer:** C

---

**68.**

```
void M(params object[] o)
M(1, "A", true);
```

Result?

- A) Error
- B) 3 arguments
- C) Boxing issue
- D) CLS error

 **Answer:** B

---

### 69. Which is executed last?

- A) Constructor
- B) Dispose
- C) Finalizer
- D) Method

 **Answer:** C

---

### 70. Which avoids memory leak?

- A) GC only
- B) Destructor only
- C) Dispose pattern
- D) CTS

 **Answer:** C

---

### 71. Which allows cleaner APIs?

- A) Positional params
- B) Named params
- C) ref
- D) out

 **Answer:** B

---

### 72.

```
class A {  
    public int X { get; }  
}
```

Assignment possible?

- A) Anywhere
- B) Constructor only
- C) Setter required
- D) Not possible

 **Answer:** B

---

### 73. Which increases coupling?

- A) Properties
- B) Public fields
- C) Object initializer
- D) Local functions

 **Answer:** B

---

### 74. Which is NOT value type?

- A) int
- B) bool
- C) struct
- D) string

 **Answer:** D

---

### 75. Which ensures deterministic cleanup?

- A) GC
- B) Finalizer
- C) Dispose
- D) Destructor

 **Answer:** C

---

### 76.

```
int Add(int x, int y=5, int z=10)
Add(1, z:20);
```

Result?

- A) 26

- A) 16
- C) Error
- D) 25

 **Answer:** B

---

### **77. Which allows chaining initialization?**

- A) Constructor
- B) Object initializer
- C) Destructor
- D) Params

 **Answer:** B

---

### **78. Which is executed per object?**

- A) Static constructor
- B) Instance constructor
- C) Destructor
- D) Finalizer

 **Answer:** B

---

### **79. Which is CLS compliant?**

- A) Pointer
- B) Overload by return type
- C) Method overloading
- D) Unsigned type

 **Answer:** C

---

### **80. Which improves performance?**

- A) Destructor
- B) Finalizer
- C) Dispose
- D) GC.Collect

 **Answer:** C

---

## **81. Which causes compile error?**

```
void M(int x, int x)
```

- A) Overload
- B) Shadowing
- C) Duplicate params
- D) CLS issue

 **Answer:** C

---

## **82. Which allows nested logic?**

- A) Local functions
- B) Properties
- C) Fields
- D) Namespaces

 **Answer:** A

---

## **83. Which is preferred?**

- A) Public fields
- B) Properties
- C) Global vars
- D) Static data

 **Answer:** B

---

## **84. Which is fastest cleanup?**

- A) GC
- B) Finalizer
- C) Dispose
- D) Destructor

 **Answer:** C

---

## **85. Which is true?**

- A) Destructor is reliable
- B) Dispose is optional

- C) GC is slow
- D) Finalizer is non-deterministic

 **Answer:** D

---

### **86. Which feature improves maintainability?**

- A) Fields
- B) Properties
- C) Params
- D) Ref

 **Answer:** B

---

### **87. Which allows flexible object creation?**

- A) Constructor
- B) Object initializer
- C) Destructor
- D) Local function

 **Answer:** B

---

### **88. Which should be avoided?**

- A) Dispose pattern
- B) Finalizers
- C) Properties
- D) Interfaces

 **Answer:** B

---

### **89. Which is CTS role?**

- A) Syntax
- B) Types definition
- C) Compilation
- D) Memory

 **Answer:** B

---

## **90. Which best practice?**

- A) GC.Collect()
- B) Dispose unmanaged resources
- C) Rely on finalizers
- D) Ignore cleanup

 **Answer:** B

# **SESSION 4 – STATIC & INHERITANCE (CCEE MCQs)**

---

## **EASY LEVEL (25 MCQs)**

### **1. Static members belong to:**

- A) Object
- B) Class
- C) Interface
- D) Namespace

 **Answer:** B

---

### **2. How many copies of a static field exist?**

- A) One per object
- B) One per method
- C) One per class
- D) One per thread

 **Answer:** C

---

### **3. Which keyword is used to declare static members?**

- A) const
- B) readonly
- C) static
- D) final

 Answer: C

---

#### 4. Static methods can access:

- A) Instance members directly
- B) Only static members
- C) Protected members
- D) Virtual members

 Answer: B

---

#### 5. Static constructor is called:

- A) Every time object is created
- B) Before Main
- C) Once per class automatically
- D) Manually

 Answer: C

---

#### 6. Static constructors:

- A) Can have parameters
- B) Can be overloaded
- C) Cannot have access modifiers
- D) Can be public

 Answer: C

---

#### 7. Static class:

- A) Can be instantiated
- B) Can have instance members
- C) Cannot be inherited
- D) Can implement interfaces

 Answer: C

---

#### 8. Which is TRUE about static class?

- A) Can have constructors
- B) Must be sealed implicitly
- C) Can be inherited
- D) Can contain instance fields

 Answer: B

---

## 9. Static local functions are:

- A) Global functions
- B) Methods inside class
- C) Functions inside a method
- D) Lambda expressions

 Answer: C

---

## 10. Inheritance represents:

- A) Has-a relationship
- B) Is-a relationship
- C) Uses-a relationship
- D) Depends-on relationship

 Answer: B

---

## 11. Which keyword is used for inheritance?

- A) implements
- B) inherits
- C) :
- D) extends

 Answer: C

---

## 12. Which access specifier allows access in derived class?

- A) private
- B) protected
- C) internal
- D) sealed

 Answer: B

---

### **13. Which access specifier is most restrictive?**

- A) public
- B) protected
- C) internal
- D) private

 **Answer:** D

---

### **14. Base class constructor is called:**

- A) After derived constructor
- B) Before derived constructor
- C) Never automatically
- D) Only manually

 **Answer:** B

---

### **15. Which keyword hides base class member?**

- A) override
- B) sealed
- C) new
- D) virtual

 **Answer:** C

---

### **16. Which keyword enables runtime polymorphism?**

- A) new
- B) override
- C) sealed
- D) static

 **Answer:** B

---

### **17. Abstract class:**

- A) Can be instantiated
- B) Must have abstract methods

- C) Cannot have concrete methods
- D) Cannot be sealed

 **Answer:** D

---

### **18. Abstract method:**

- A) Has body
- B) Is private
- C) Must be overridden
- D) Is static

 **Answer:** C

---

### **19. Sealed class:**

- A) Can be inherited
- B) Cannot be inherited
- C) Must be abstract
- D) Must be static

 **Answer:** B

---

### **20. Sealed method:**

- A) Cannot be overridden further
- B) Cannot be inherited
- C) Cannot be called
- D) Cannot be virtual

 **Answer:** A

---

### **21. Which cannot be static?**

- A) Field
- B) Method
- C) Constructor
- D) Local function

 **Answer:** C

---

## **22. Which method is resolved at compile time?**

- A) virtual
- B) override
- C) static
- D) abstract

 **Answer:** C

---

## **23. Which supports polymorphism?**

- A) static method
- B) sealed method
- C) virtual method
- D) private method

 **Answer:** C

---

## **24. Which keyword prevents inheritance?**

- A) abstract
- B) sealed
- C) private
- D) static

 **Answer:** B

---

## **25. Which class cannot contain abstract methods?**

- A) Abstract class
- B) Sealed class
- C) Base class
- D) Derived class

 **Answer:** B

---

## **🟡 MEDIUM LEVEL (25 MCQs – MANY SNIPPETS)**

### **26.**

```
class A {  
    static int x = 10;
```

}

x belongs to?

- A) Object
- B) Method
- C) Class
- D) Thread

 Answer: C

---

**27.**

```
static class A {  
    public int x;  
}
```

Result?

- A) Valid
- B) Runtime error
- C) Compile error
- D) Warning

 Answer: C

---

**28.**

```
class A {  
    static A(){}  
}
```

Which is true?

- A) Overloaded
- B) Called once
- C) Needs call
- D) Has parameters

 Answer: B

---

**29.**

```
class A {  
    public static int X { get; set; }
```

}

Property type?

- A) Instance
- B) Static
- C) Read-only
- D) Virtual

 Answer: B

---

30.

```
class A {  
    public void M(){}  
}  
class B:A {  
    public void M(){}  
}
```

This is:

- A) Override
- B) Overload
- C) Method hiding
- D) Error

 Answer: C

---

31.

```
class A {  
    public virtual void M(){}  
}  
class B:A {  
    public override void M(){}  
}
```

Type of binding?

- A) Compile-time
- B) Runtime
- C) Static
- D) Early

 Answer: B

---

**32.**

```
class A {  
    public void M(){}  
}  
class B:A {  
    public new void M(){}  
}
```

Which keyword hides?

- A) override
- B) sealed
- C) new
- D) static

 **Answer:** C

---

**33.**

```
abstract class A {  
    public void M(){}  
}
```

Valid?

- A) Yes
- B) No
- C) Warning
- D) Runtime error

 **Answer:** A

---

**34.**

```
abstract class A {  
    public abstract void M();  
}  
class B:A {}
```

Result?

- A) Valid
- B) Compile error

- C) Runtime error
- D) Warning

 **Answer:** B

---

**35.**

```
sealed class A {}  
class B:A {}
```

Result?

- A) Valid
- B) Runtime error
- C) Compile error
- D) Warning

 **Answer:** C

---

**36.**

```
class A {  
    protected int x;  
}  
class B:A {  
    void M(){ x=10; }  
}
```

Result?

- A) Error
- B) Valid
- C) Warning
- D) Runtime error

 **Answer:** B

---

**37. Which access specifier is inherited?**

- A) private
- B) protected
- C) sealed
- D) static

 **Answer:** B

---

**38.**

```
class A {  
    public A(){}  
}  
class B:A {  
    public B(){}  
}
```

Execution order?

- A) B → A
- B) A → B
- C) Only B
- D) Only A

 **Answer:** B

---

**39. Which member cannot be overridden?**

- A) virtual
- B) abstract
- C) sealed
- D) protected

 **Answer:** C

---

**40. Which hides base implementation?**

- A) override
- B) new
- C) sealed
- D) static

 **Answer:** B

---

**41.**

```
class A {  
    public static void M(){}  
}  
class B:A {}
```

Can B override M?

- A) Yes
- B) No
- C) Runtime error
- D) Warning

 Answer: B

---

#### 42. Which supports runtime polymorphism?

- A) new
- B) static
- C) override
- D) sealed

 Answer: C

---

#### 43.

```
abstract class A {}  
A a = new A();
```

Result?

- A) Valid
- B) Compile error
- C) Runtime error
- D) Warning

 Answer: B

---

#### 44. Which must be overridden?

- A) virtual
- B) abstract
- C) static
- D) sealed

 Answer: B

---

#### 45.

```
class A {  
    static int x;  
    static A(){ x=5; }  
}
```

When does x initialize?

- A) On object creation
- B) On first access
- C) On every call
- D) On GC

 Answer: B

---

#### 46. Which is TRUE?

- A) Static methods can be virtual
- B) Static classes can be inherited
- C) Static constructors run once
- D) Static fields are per object

 Answer: C

---

#### 47. Which breaks polymorphism?

- A) override
- B) abstract
- C) sealed method
- D) virtual

 Answer: C

---

#### 48.

```
class A {  
    public virtual void M(){}
}  
class B:A {  
    public sealed override void M(){}
}
```

Which is true?

- A) Can override again

- B) Compile error
- C) Cannot override further
- D) Hides method

 Answer: C

---

#### 49. Which is best practice?

- A) Public fields
- B) Protected fields
- C) Encapsulation via properties
- D) Static everywhere

 Answer: C

---

#### 50. Which is resolved at runtime?

- A) static call
- B) new method
- C) override method
- D) constructor

 Answer: C

---

### HARD LEVEL (25 MCQs – TRICKY + SNIPPETS)

#### 51.

```
class A {  
    public virtual void M(){}  
}  
class B:A {  
    public new void M(){}  
}  
A a = new B();  
a.M();
```

Which method is called?

- A) A.M
- B) B.M
- C) Error
- D) Depends

**Answer:** A

---

**52.**

```
class A {  
    public static int x=10;  
}  
class B:A {}  
Console.WriteLine(B.x);
```

Output?

- A) Error
- B) 0
- C) 10
- D) Undefined

**Answer:** C

---

**53.**

```
abstract class A {  
    public abstract void M();  
}  
abstract class B:A {}
```

Result?

- A) Error
- B) Valid
- C) Runtime error
- D) Warning

**Answer:** B

---

**54.**

```
sealed class A {  
    public virtual void M(){}  
}
```

Result?

- A) Valid
- B) Compile error

- C) Runtime error
- D) Warning

 **Answer:** A

---

### 55.

```
class A {  
    protected A(){}  
}  
class B:A {  
    public B(){}  
}
```

Result?

- A) Valid
- B) Error
- C) Warning
- D) Runtime error

 **Answer:** A

---

### 56. Which causes compile error?

- A) Sealed override
- B) Override non-virtual
- C) Abstract method
- D) Static constructor

 **Answer:** B

---

### 57.

```
static class A {  
    static A(){}  
}
```

Constructor called?

- A) Manually
- B) On first access
- C) On object creation
- D) On GC

**Answer:** B

---

### **58. Which is FALSE?**

- A) Static methods cannot be overridden
- B) new hides base member
- C) sealed method allows override
- D) abstract class can have fields

**Answer:** C

---

### **59.**

```
class A {  
    public virtual void M(){}  
}  
class B:A {  
    public override void M(){}  
}  
class C:B {}
```

Can C override M?

- A) Yes
- B) No
- C) Runtime error
- D) Warning

**Answer:** A

---

### **60. Which ensures safety?**

- A) Public inheritance
- B) Sealed class
- C) new keyword
- D) Static fields

**Answer:** B

---

### **61. Which is NOT inherited?**

- A) protected
- B) public
- C) private
- D) internal

 **Answer:** C

---

## 62.

```
class A {  
    public virtual void M(){}  
}  
class B:A {  
    public override void M(){}  
}  
class C:B {  
    public new void M(){}  
}
```

C.M is?

- A) Override
- B) Hiding
- C) Error
- D) Sealed

 **Answer:** B

---

## 63. Which improves performance?

- A) override
- B) virtual
- C) static
- D) abstract

 **Answer:** C

---

## 64. Which breaks inheritance?

- A) sealed class
- B) abstract class
- C) virtual method
- D) protected field

 **Answer:** A

---

**65.**

```
class A {  
    public virtual void M(){}
}  
class B:A {  
    public sealed override void M(){}
}
```

What happens?

- A) Can override again
- B) Compile error
- C) Prevents further override
- D) Hides method

 **Answer:** C

---

**66. Which keyword prevents accidental hiding?**

- A) override
- B) sealed
- C) static
- D) protected

 **Answer:** A

---

**67. Which is compile-time polymorphism?**

- A) override
- B) virtual
- C) method overloading
- D) inheritance

 **Answer:** C

---

**68. Which feature supports abstraction?**

- A) static
- B) sealed

- C) abstract
- D) new

 **Answer:** C

---

### **69. Which is recommended?**

- A) Hide methods
- B) Use override keyword
- C) Use public fields
- D) Avoid base constructors

 **Answer:** B

---

### **70. Which avoids misuse?**

- A) public class
- B) sealed class
- C) virtual method
- D) protected method

 **Answer:** B

---

### **71.**

```
class A {  
    public virtual void M(){}  
}  
class B:A {  
    public override void M(){}  
}  
A a = new A();  
a.M();
```

Called?

- A) A.M
- B) B.M
- C) Error
- D) Depends

 **Answer:** A

---

## **72. Which is CLS compliant?**

- A) Overload by return type
- B) Method overloading
- C) Pointer usage
- D) Unsigned types

 **Answer:** B

---

## **73. Which design is safer?**

- A) Public fields
- B) Protected fields
- C) Private + properties
- D) Static globals

 **Answer:** C

---

## **74. Which improves maintainability?**

- A) new keyword
- B) override keyword
- C) public fields
- D) sealed everywhere

 **Answer:** B

---

## **75. Which is BEST practice?**

- A) Avoid inheritance
- B) Prefer composition
- C) Use sealed when needed
- D) All of the above

 **Answer:** D

 **SESSION 5 & 6 – Interfaces, Types, Arrays  
(CCEE MCQs)**

---

## EASY LEVEL (25 MCQs)

### 1. An interface in C# contains:

- A) Fields
- B) Constructors
- C) Method declarations
- D) Static variables

 Answer: C

---

### 2. Which keyword is used to implement an interface?

- A) implements
- B) extends
- C) :
- D) inherit

 Answer: C

---

### 3. Interfaces support:

- A) Multiple inheritance
- B) Single inheritance
- C) No inheritance
- D) Constructor inheritance

 Answer: A

---

### 4. Which member is allowed in an interface (C# 8+)?

- A) Fields
- B) Constructors
- C) Default methods
- D) Destructors

 Answer: C

---

### 5. Which keyword is used for explicit interface implementation?

- A) explicit
- B) override
- C) InterfaceName.Method
- D) sealed

 **Answer:** C

---

## **6. Which interface is used to release unmanaged resources?**

- A) ICloneable
- B) IComparable
- C) IDisposable
- D) IEnumerable

 **Answer:** C

---

## **7. Which interface is used for object comparison?**

- A) IComparer
- B) IComparable
- C) IDisposable
- D) IEnumerator

 **Answer:** B

---

## **8. struct is a:**

- A) Reference type
- B) Value type
- C) Pointer type
- D) Nullable type

 **Answer:** B

---

## **9. enum is used to represent:**

- A) Objects
- B) Constants
- C) Strings
- D) Classes

 **Answer:** B

---

## **10. Which is a value type?**

- A) class
- B) array
- C) struct
- D) string

 **Answer:** C

---

## **11. ref parameter:**

- A) Passes copy
- B) Passes reference
- C) Is output only
- D) Is optional

 **Answer:** B

---

## **12. out parameter:**

- A) Must be initialized before call
- B) Must be assigned inside method
- C) Cannot return value
- D) Is optional

 **Answer:** B

---

## **13. Nullable value type syntax?**

- A) int\*
- B) int?
- C) int!
- D) Nullable<int\*>

 **Answer:** B

---

## **14. ?? operator is called:**

- A) Conditional
- B) Ternary

- C) Null-coalescing
- D) Range

 Answer: C

---

### 15. ??= operator does:

- A) Comparison
- B) Assignment if null
- C) Increment
- D) Cast

 Answer: B

---

### 16. Which array has rows of different sizes?

- A) Single-dimensional
- B) Multi-dimensional
- C) Jagged
- D) Static

 Answer: C

---

### 17. Indexers allow objects to be accessed like:

- A) Methods
- B) Fields
- C) Arrays
- D) Properties

 Answer: C

---

### 18. Index operator is represented by:

- A) ..
- B) :
- C) ^
- D) []

 Answer: D

---

### **19. ^1 index refers to:**

- A) First element
- B) Last element
- C) Second element
- D) Middle element

 **Answer:** B

---

### **20. Ranges are defined using:**

- A) :
- B) ..
- C) ,
- D) =>

 **Answer:** B

---

### **21. Which is reference type?**

- A) struct
- B) enum
- C) class
- D) int

 **Answer:** C

---

### **22. Arrays are:**

- A) Value types
- B) Reference types
- C) Pointer types
- D) Nullable types

 **Answer:** B

---

### **23. Which interface supports foreach?**

- A) IDisposable
- B) IComparable
- C) IEnumerable
- D) ICloneable

 Answer: C

---

#### 24. Default interface methods were introduced in:

- A) C# 6
- B) C# 7
- C) C# 8
- D) C# 9

 Answer: C

---

#### 25. Which improves type safety?

- A) object
- B) nullable reference types
- C) boxing
- D) casting

 Answer: B

---

### MEDIUM LEVEL (25 MCQs – MANY SNIPPETS)

#### 26.

```
interface ITest {  
    void M();  
}  
class A : ITest {}
```

Result?

- A) Valid
- B) Compile error
- C) Runtime error
- D) Warning

 Answer: B

---

#### 27.

```
interface I {  
    void M();
```

```
}
```

```
class A : I {
```

```
    void I.M(){}
}
```

This is:

- A) Override
- B) Explicit implementation
- C) Method hiding
- D) Error

 **Answer:** B

---

**28.**

```
IComparable x;
```

Purpose?

- A) Dispose
- B) Compare objects
- C) Clone object
- D) Enumerate

 **Answer:** B

---

**29.**

```
struct S {
```

```
    public int x;
```

```
}
```

Where is S stored?

- A) Heap
- B) Stack (generally)
- C) LOH
- D) Metadata

 **Answer:** B

---

**30.**

```
int? x = null;
```

```
Console.WriteLine(x ?? 5);
```

Output?

- A) null
- B) 0
- C) 5
- D) Error

 **Answer:** C

---

**31.**

```
string? s = null;  
Console.WriteLine(s.Length);
```

Result?

- A) 0
- B) Runtime error
- C) Compile warning/error
- D) null

 **Answer:** C

---

**32.**

```
void M(ref int x){ x++; }
```

Requirement?

- A) x initialized before call
- B) x assigned inside method
- C) optional
- D) readonly

 **Answer:** A

---

**33.**

```
void M(out int x){ x=10; }
```

Which is true?

- A) x must be initialized before
- B) x must be assigned inside

- C) x is optional
- D) Error

 **Answer:** B

---

**34.**

```
int[] a = {1,2,3,4};  
Console.WriteLine(a[1]);
```

Output?

- A) 1
- B) 2
- C) 3
- D) 4

 **Answer:** D

---

**35.**

```
int[] a = {1,2,3,4,5};  
var b = a[1..4];
```

b contains?

- A) 2,3,4
- B) 1,2,3
- C) 2,3,4,5
- D) Error

 **Answer:** A

---

**36.**

```
interface A{}  
interface B : A{}
```

This is:

- A) Invalid
- B) Interface inheritance
- C) Class inheritance
- D) Error

 **Answer:** B

---

**37.**

```
int[,] a = new int[2,3];
```

Type?

- A) Jagged
- B) Single
- C) Multi-dimensional
- D) Dynamic

 **Answer:** C

---

**38.**

```
int[][] a = new int[2][];
```

Type?

- A) Jagged
- B) Multi-dimensional
- C) Static
- D) Invalid

 **Answer:** A

---

**39.**

```
class A : IDisposable {}
```

Missing?

- A) Constructor
- B) Finalizer
- C) Dispose method
- D) Destructor

 **Answer:** C

---

**40.**

```
public int this[int i] { get; set; }
```

This defines?

- A) Property
- B) Method
- C) Indexer
- D) Field

 Answer: C

---

#### 41. Which is boxing?

- A) int → object
- B) object → int
- C) string → char
- D) class → struct

 Answer: A

---

#### 42. Which avoids NullReferenceException?

- A) ??
- B) ref
- C) out
- D) enum

 Answer: A

---

#### 43. Which allows array-like access to class?

- A) Properties
- B) Fields
- C) Indexers
- D) Interfaces

 Answer: C

---

#### 44. Which improves performance?

- A) struct over class
- B) class over struct
- C) boxing
- D) interface calls

 Answer: A

---

**45.**

`Employee[ ] e = new Employee[5];`

Type?

- A) Value type array
- B) Reference type array
- C) Stack allocated
- D) Nullable

 **Answer:** B

---

**46. Which interface supports sorting?**

- A) IDisposable
- B) IComparable
- C) IEnumerable
- D) ICloneable

 **Answer:** B

---

**47. Which supports polymorphism?**

- A) struct
- B) enum
- C) interface
- D) array

 **Answer:** C

---

**48. Which is FALSE?**

- A) struct supports inheritance from interface
- B) enum is value type
- C) interface can have fields
- D) arrays are reference types

 **Answer:** C

---

**49.**

```
x ??= 10;
```

Means?

- A) Assign 10 always
- B) Assign 10 if x is null
- C) Compare with 10
- D) Error

 Answer: B

---

## 50. Which requires explicit implementation?

- A) Same method names
- B) Conflicting interface methods
- C) Abstract methods
- D) Virtual methods

 Answer: B

---

## HARD LEVEL (25 MCQs – TRICKY + SNIPPETS)

### 51.

```
interface I {  
    void M();  
}  
class A : I {  
    public void M(){}  
}  
I i = new A();  
i.M();
```

Call resolved at?

- A) Compile time
- B) Runtime
- C) Link time
- D) JIT

 Answer: B

---

### 52.

```
struct S { public int x; }
S s1 = new S();
S s2 = s1;
s2.x = 10;
```

s1.x?

- A) 10
- B) 0
- C) null
- D) Error

 **Answer:** B

---

**53.**

```
int? x = null;
Console.WriteLine(x.Value);
```

Result?

- A) 0
- B) null
- C) Runtime exception
- D) Compile error

 **Answer:** C

---

**54. Which breaks CLS?**

- A) Interfaces
- B) Nullable types
- C) Unsigned types
- D) Indexers

 **Answer:** C

---

**55.**

```
class A {
    public int this[int i] => i*2;
}
Console.WriteLine(new A()[3]);
```

Output?

- A) 3
- B) 6
- C) Error
- D) 0

 Answer: B

---

### 56. Which is NOT true?

- A) struct is copied by value
- B) interface supports fields
- C) arrays are reference types
- D) enum is value type

 Answer: B

---

### 57. Which avoids boxing?

- A) interface usage
- B) generics
- C) object reference
- D) casting

 Answer: B

---

### 58.

```
void M(out int x){}
```

Result?

- A) Valid
- B) Compile error
- C) Runtime error
- D) Warning

 Answer: B

---

### 59.

```
int[] a = null;  
Console.WriteLine(a?.Length ?? 0);
```

Output?

- A) Error
- B) null
- C) 0
- D) Exception

 **Answer:** C

---

## 60. Which is safer?

- A) nullable value types
- B) nullable reference types
- C) object
- D) dynamic

 **Answer:** B

---

## 61. Which is resolved at runtime?

- A) struct copy
- B) interface call
- C) enum value
- D) array indexing

 **Answer:** B

---

## 62.

```
interface I1 { void M(); }
interface I2 { void M(); }
class A : I1, I2 {
    void I1.M(){}
    void I2.M(){}
}
```

This is:

- A) Overload
- B) Explicit interface implementation
- C) Error
- D) Override

 **Answer:** B

---

### **63. Which improves memory locality?**

- A) class
- B) struct
- C) interface
- D) array of objects

 **Answer:** B

---

### **64. Which is true?**

- A) Arrays are value types
- B) struct supports inheritance from class
- C) interface supports multiple inheritance
- D) enum can have methods

 **Answer:** C

---

### **65. Which is slowest?**

- A) struct call
- B) class call
- C) interface call
- D) static call

 **Answer:** C

---

### **66. Which avoids NullReferenceException best?**

- A) if checks
- B) ?? operator
- C) try-catch
- D) GC

 **Answer:** B

---

### **67. Which cannot be nullable?**

- A) int
- B) struct
- C) enum
- D) void

 **Answer:** D

---

## 68. Which is compile-time safe?

- A) dynamic
- B) nullable references
- C) object
- D) boxing

 **Answer:** B

---

## 69. Which is recommended?

- A) Use class everywhere
- B) Use struct for small immutable data
- C) Avoid interfaces
- D) Avoid arrays

 **Answer:** B

---

## 70. Which improves maintainability?

- A) Explicit interface implementation
- B) Public fields
- C) Boxing
- D) Static globals

 **Answer:** A

---

## 71.

```
int[] a = {1,2,3};  
var b = a[..^1];
```

b contains?

- A) 1,2
- B) 2,3

- C) 1,2,3
- D) Error

 **Answer:** A

---

## 72. Which is BEST practice?

- A) Avoid IDisposable
- B) Implement IDisposable for unmanaged resources
- C) Rely on GC only
- D) Use finalizers

 **Answer:** B

---

## 73. Which enables contract-based design?

- A) class
- B) struct
- C) interface
- D) enum

 **Answer:** C

---

## 74. Which reduces memory pressure?

- A) class
- B) struct
- C) boxing
- D) interface

 **Answer:** B

---

## 75. Which is MOST important for CCEE?

- A) Syntax
- B) Concept clarity
- C) Framework internals
- D) UI

 **Answer:** B

# SESSION 7 – Generics, Collections, Tuples (CCEE MCQs)

---

## EASY LEVEL (25 MCQs)

### 1. Generics provide:

- A) Runtime type safety
- B) Compile-time type safety
- C) Faster IO
- D) UI support

 Answer: B

---

### 2. Which symbol is used to define generics?

- A) ()
- B) []
- C) < >
- D) {}

 Answer: C

---

### 3. Which is a generic collection?

- A) ArrayList
- B) Hashtable
- C) List<T>
- D) Queue

 Answer: C

---

### 4. Which namespace contains generic collections?

- A) System
- B) System.IO
- C) System.Collections
- D) System.Collections.Generic

 Answer: D

---

## **5. Which is a non-generic collection?**

- A) List<int>
- B) Dictionary<int,string>
- C) ArrayList
- D) HashSet<int>

 **Answer:** C

---

## **6. ICollection<T> provides:**

- A) Index access
- B) Add/Remove functionality
- C) Key-value pairs
- D) Sorting

 **Answer:** B

---

## **7. IList<T> supports:**

- A) Enumeration only
- B) Index-based access
- C) Key lookup
- D) Thread safety

 **Answer:** B

---

## **8. IDictionary< TKey, TValue > stores data as:**

- A) Values
- B) Indexes
- C) Objects
- D) Key-value pairs

 **Answer:** D

---

## **9. Which loop is best for collections?**

- A) for
- B) while

- A) foreach
- B) do-while

 **Answer:** C

---

## 10. Generic collections avoid:

- A) Polymorphism
- B) Boxing and unboxing
- C) Interfaces
- D) Iteration

 **Answer:** B

---

## 11. Which collection allows duplicate values?

- A) HashSet
- B) Dictionary
- C) List
- D) SortedDictionary

 **Answer:** C

---

## 12. Which interface supports foreach?

- A) IDisposable
- B) IEnumerator
- C) IEnumerable
- D) IComparable

 **Answer:** C

---

## 13. Which tuple type is value type?

- A) Tuple<T>
- B) ValueTuple<T>
- C) object
- D) dynamic

 **Answer:** B

---

## **14. Which tuple syntax is correct?**

- A) Tuple<int,int>
- B) (int,int)
- C) <int,int>
- D) [int,int]

 **Answer:** B

---

## **15. Generic methods allow:**

- A) Multiple return values
- B) Type-safe reuse
- C) Method overriding
- D) Runtime casting

 **Answer:** B

---

## **16. Which is true?**

- A) Generics are slower
- B) Generics work at runtime only
- C) Generics improve performance
- D) Generics require reflection

 **Answer:** C

---

## **17. Which interface is implemented by List<T>?**

- A) IDictionary
- B) IList
- C) IComparer
- D) IDisposable

 **Answer:** B

---

## **18. Which collection stores unique values?**

- A) List
- B) ArrayList
- C) HashSet
- D) Queue

 Answer: C

---

### 19. Which tuple member access is valid?

- A) Item0
- B) Item1
- C) Value1
- D) Index1

 Answer: B

---

### 20. Which generic constraint restricts to reference types?

- A) where T : new()
- B) where T : struct
- C) where T : class
- D) where T : base

 Answer: C

---

### 21. Which restricts to value types?

- A) class
- B) struct
- C) new()
- D) unmanaged

 Answer: B

---

### 22. Which collection maps keys to values?

- A) List
- B) Queue
- C) Dictionary
- D) Stack

 Answer: C

---

### 23. Which non-generic collection stores objects?

- A) List
- B) ArrayList
- C) Dictionary
- D) HashSet

 **Answer:** B

---

## **24. Which allows fast lookup by key?**

- A) List
- B) Stack
- C) Dictionary
- D) Queue

 **Answer:** C

---

## **25. Which is preferred?**

- A) Non-generic collections
- B) Generic collections
- C) Arrays only
- D) Dynamic types

 **Answer:** B

---

## **MEDIUM LEVEL (25 MCQs – MANY SNIPPETS)**

### **26.**

```
List<int> l = new List<int>();  
l.Add(10);
```

What is stored?

- A) object
- B) int
- C) boxed int
- D) dynamic

 **Answer:** B

---

### **27.**

```
ArrayList a = new ArrayList();
a.Add(10);
```

10 is stored as:

- A) int
- B) boxed int
- C) reference
- D) pointer

 **Answer:** B

---

**28.**

```
void M<T>(T x) { }
```

This is:

- A) Generic class
- B) Generic method
- C) Constraint
- D) Interface

 **Answer:** B

---

**29.**

```
class Test<T> where T : class
```

Constraint?

- A) Value type
- B) Reference type
- C) Interface
- D) Constructor

 **Answer:** B

---

**30.**

```
Dictionary<int, string> d = new();
d[1] = "A";
```

Stored as?

- A) Value only

- B) Index-value
- C) Key-value
- D) Tuple

 **Answer:** C

---

**31.**

```
foreach(var x in list) { }
```

Requires?

- A) IList
- B) ICollection
- C) IEnumerable
- D) IComparable

 **Answer:** C

---

**32.**

```
HashSet<int> h = new();
h.Add(1); h.Add(1);
```

Count?

- A) 0
- B) 1
- C) 2
- D) Error

 **Answer:** B

---

**33.**

```
List<object> l = new();
l.Add(10);
```

10 is:

- A) boxed
- B) generic
- C) error
- D) value type

 **Answer:** A

---

**34.**

```
var t = (1, "A");
```

Type?

- A) Tuple
- B) ValueTuple
- C) object
- D) dynamic

 **Answer:** B

---

**35.**

```
var (x,y) = (10,20);
```

This is:

- A) Casting
- B) Deconstruction
- C) Boxing
- D) Assignment

 **Answer:** B

---

**36.**

```
class A<T> where T : new()
```

Constraint ensures?

- A) Reference type
- B) Value type
- C) Parameterless constructor
- D) Interface

 **Answer:** C

---

**37.**

```
IList<int> l = new List<int>();
```

Allowed?

- A) Yes
- B) No
- C) Runtime error
- D) Warning

 **Answer:** A

---

**38.**

```
Dictionary<int, string> d;  
d.Add(1, "A");  
d.Add(1, "B");
```

Result?

- A) Overwrites
- B) Runtime exception
- C) Compile error
- D) Warning

 **Answer:** B

---

**39.**

```
List<int> l = new();  
l.Remove(5);
```

If 5 not present?

- A) Exception
- B) false returned
- C) true returned
- D) Error

 **Answer:** B

---

**40.**

```
Tuple<int, int> t = new Tuple<int, int>(1, 2);
```

Access first?

- A) t[0]
- B) t.First

- C) t.Item1
- D) t.Value1

 **Answer:** C

---

#### **41. Which avoids boxing?**

- A) ArrayList
- B) List<int>
- C) object
- D) dynamic

 **Answer:** B

---

#### **42.**

```
void M<T>(T x) where T : struct
```

Restricts to?

- A) Class
- B) Value types
- C) Interfaces
- D) Reference types

 **Answer:** B

---

#### **43.**

```
ICollection<int> c = new List<int>();
```

Which operation allowed?

- A) Indexing
- B) Sorting
- C) Add
- D) Key lookup

 **Answer:** C

---

#### **44. Which collection maintains insertion order?**

- A) Dictionary
- B) HashSet

- C) List
- D) SortedDictionary

 **Answer:** C

---

#### 45.

`SortedDictionary<int, string>`

Sorted by?

- A) Value
- B) Key
- C) Index
- D) Insertion order

 **Answer:** B

---

#### 46. Which tuple feature improves readability?

- A) Item1
- B) Named elements
- C) Boxing
- D) Casting

 **Answer:** B

---

#### 47.

```
var t = (id:1, name:"A");
Console.WriteLine(t.name);
```

Output?

- A) 1
- B) A
- C) Error
- D) null

 **Answer:** B

---

#### 48. Which is fastest lookup?

- A) List
- B) Dictionary
- C) Stack
- D) Queue

 **Answer:** B

---

#### **49. Which is NOT generic?**

- A) Queue<T>
- B) Stack<T>
- C) Hashtable
- D) Dictionary< TKey, TValue >

 **Answer:** C

---

#### **50. Which is CLS compliant?**

- A) Tuple
- B) ValueTuple
- C) Generics
- D) unsafe code

 **Answer:** C

---

### **HARD LEVEL (25 MCQs – TRICKY + SNIPPETS)**

#### **51.**

```
List<int> l = new();  
IEnumerable<int> e = l;
```

Which operations allowed on e?

- A) Add
- B) Remove
- C) Iterate only
- D) Index

 **Answer:** C

---

#### **52.**

```
class A<T> where T : class, new()
```

Requires?

- A) Value type
- B) Reference type + default ctor
- C) Interface
- D) Struct

 **Answer:** B

---

**53.**

```
ArrayList a = new();
a.Add(1);
int x = (int)a[0];
```

This is:

- A) Generic cast
- B) Boxing + unboxing
- C) Error
- D) No cast

 **Answer:** B

---

**54.**

```
List<int> l = new();
l.AddRange(new int[]{1,2});
```

Valid?

- A) Yes
- B) No
- C) Runtime error
- D) Warning

 **Answer:** A

---

**55.**

```
Dictionary<int,string> d = new();
Console.WriteLine(d[1]);
```

If key missing?

- A) null
- B) ""
- C) Exception
- D) 0

 Answer: C

---

## 56. Which breaks type safety?

- A) Generics
- B) Tuple
- C) ArrayList
- D) Dictionary

 Answer: C

---

## 57. Which is fastest iteration?

- A) foreach on List
- B) foreach on ArrayList
- C) foreach on Dictionary
- D) foreach on Stack

 Answer: A

---

## 58.

(T1 , T2) M<T1 , T2>(T1 a , T2 b)

Returns?

- A) Class
- B) Tuple
- C) ValueTuple
- D) Array

 Answer: C

---

## 59.

```
var t = (1,2);  
t.Item1 = 10;
```

Result?

- A) Valid
- B) Compile error
- C) Runtime error
- D) Warning

 Answer: A

---

## 60. Which is immutable?

- A) List
- B) Dictionary
- C) ValueTuple
- D) Array

 Answer: C

---

## 61. Which constraint is checked at compile time?

- A) new()
- B) class
- C) struct
- D) All

 Answer: D

---

## 62.

```
class Test<T> where T : IComparable
```

Allows?

- A) Comparison
- B) Disposal
- C) Cloning
- D) Enumeration

 Answer: A

---

## 63. Which avoids runtime errors?

- A) Non-generic
- B) Generic collections
- C) Casting
- D) dynamic

 **Answer:** B

---

#### **64.**

```
foreach(var x in dict)
```

x is?

- A) Key
- B) Value
- C) KeyValuePair
- D) Tuple

 **Answer:** C

---

#### **65. Which supports index access?**

- A) ICollection
- B) IEnumerable
- C) IList
- D) IDictionary

 **Answer:** C

---

#### **66. Which is slowest?**

- A) List<int>
- B) ArrayList
- C) Dictionary<int,int>
- D) Stack<int>

 **Answer:** B

---

#### **67. Which is BEST practice?**

- A) Use ArrayList
- B) Use List<T>

- C) Use object
- D) Use dynamic

 **Answer:** B

---

## 68.

```
List<string> l = new();  
l.Add(null);
```

Allowed?

- A) Yes
- B) No
- C) Runtime error
- D) Warning

 **Answer:** A

---

## 69. Which is reference type?

- A) ValueTuple
- B) Tuple
- C) int
- D) enum

 **Answer:** B

---

## 70. Which supports multiple return values cleanly?

- A) out parameters
- B) Tuple
- C) ArrayList
- D) Dictionary

 **Answer:** B

---

## 71.

```
IEnumerable<int> e = new List<int>();
```

Why use this?

- A) Performance

- A) Abstraction
- B) Sorting
- C) Boxing

 **Answer:** B

---

## 72. Which improves API design?

- A) Tuples
- B) ArrayList
- C) object
- D) dynamic

 **Answer:** A

---

## 73. Which ensures compile-time safety?

- A) dynamic
- B) object
- C) Generics
- D) Casting

 **Answer:** C

---

## 74. Which is NOT true?

- A) Generics reduce runtime errors
- B) ArrayList causes boxing
- C) Dictionary allows duplicate keys
- D) List allows duplicate values

 **Answer:** C

---

## 75. Which is MOST asked in CCEE?

- A) UI controls
- B) Generic vs non-generic
- C) Reflection
- D) Unsafe code

 **Answer:** B

# SESSION 8 – Delegates, Anonymous Methods, Lambdas (CCEE MCQs)

---

## EASY LEVEL (25 MCQs)

### 1. A delegate is:

- A) A class
- B) A reference type that holds method reference
- C) A value type
- D) An interface

 Answer: B

---

### 2. Delegates are similar to:

- A) Pointers to data
- B) Function pointers
- C) Classes
- D) Interfaces

 Answer: B

---

### 3. Which keyword defines a delegate?

- A) delegate
- B) event
- C) func
- D) action

 Answer: A

---

### 4. Delegates provide support for:

- A) Encapsulation
- B) Inheritance

- C) Callbacks
- D) Polymorphism only

 **Answer:** C

---

## **5. Which delegate returns void?**

- A) Func
- B) Predicate
- C) Action
- D) Event

 **Answer:** C

---

## **6. Which delegate returns a value?**

- A) Action
- B) Func
- C) Predicate
- D) Event

 **Answer:** B

---

## **7. Predicate<T> returns:**

- A) int
- B) void
- C) bool
- D) object

 **Answer:** C

---

## **8. Which allows multiple methods to be called?**

- A) Single delegate
- B) Multicast delegate
- C) Generic delegate
- D) Static delegate

 **Answer:** B

---

## **9. Anonymous methods use which keyword?**

- A) lambda
- B) delegate
- C) func
- D) action

 **Answer:** B

---

## **10. Lambda expressions use:**

- A) =>
- B) ->
- C) ::
- D) ==

 **Answer:** A

---

## **11. Delegates are:**

- A) Value types
- B) Reference types
- C) Pointer types
- D) Nullable types

 **Answer:** B

---

## **12. Which namespace contains Action and Func?**

- A) System
- B) System.Linq
- C) System.Collections
- D) System.Delegates

 **Answer:** A

---

## **13. Which delegate is predefined?**

- A) Custom delegate
- B) Action
- C) EventHandler
- D) Both B and C

 Answer: D

---

#### **14. Which feature improves code readability?**

- A) Anonymous methods
- B) Lambdas
- C) Multicast delegates
- D) Events

 Answer: B

---

#### **15. Delegates enable:**

- A) Runtime method binding
- B) Compile-time inheritance
- C) Memory management
- D) Garbage collection

 Answer: A

---

#### **16. Which can be assigned to a delegate?**

- A) Variable
- B) Property
- C) Method with matching signature
- D) Class

 Answer: C

---

#### **17. Lambda expressions are:**

- A) Classes
- B) Inline functions
- C) Interfaces
- D) Structs

 Answer: B

---

#### **18. Which delegate supports only one parameter and bool return?**

- A) Func
- B) Action
- C) Predicate
- D) EventHandler

 Answer: C

---

### 19. Multicast delegates internally use:

- A) Stack
- B) Queue
- C) Linked list
- D) Invocation list

 Answer: D

---

### 20. Which operator combines delegates?

- A) &
- B) +
- C) \*
- D) /

 Answer: B

---

### 21. Which removes a method from delegate?

- A) +
- B) -
- C) delete
- D) clear

 Answer: B

---

### 22. Delegates are sealed by default:

- A) True
- B) False

 Answer: A

---

### **23. Which delegate is best for filters?**

- A) Action
- B) Func
- C) Predicate
- D) EventHandler

 **Answer:** C

---

### **24. Which supports functional programming?**

- A) Delegates
- B) Lambdas
- C) Anonymous methods
- D) All of the above

 **Answer:** D

---

### **25. Which is preferred in modern C#?**

- A) Anonymous methods
- B) Lambdas
- C) Method pointers
- D) Function names

 **Answer:** B

---

## **MEDIUM LEVEL (25 MCQs – MANY SNIPPETS)**

### **26.**

```
delegate void D();
void M(){ Console.WriteLine("A"); }
D d = M;
d();
```

Output?

- A) A
- B) M
- C) Error
- D) Nothing

 **Answer:** A

---

**27.**

```
Action a = () => Console.WriteLine("X");  
a();
```

Output?

- A) X
- B) Error
- C) Nothing
- D) ()

 **Answer:** A

---

**28.**

```
Func<int,int> f = x => x * 2;  
Console.WriteLine(f(3));
```

Output?

- A) 3
- B) 5
- C) 6
- D) Error

 **Answer:** C

---

**29.**

```
Predicate<int> p = x => x > 5;  
Console.WriteLine(p(4));
```

Output?

- A) true
- B) false
- C) 0
- D) Error

 **Answer:** B

---

**30.**

```
Action<int> a = x => Console.WriteLine(x);  
a(10);
```

Output?

- A) x
- B) 10
- C) Error
- D) Nothing

 **Answer:** B

---

**31.**

```
Action a = delegate { Console.WriteLine("Hi"); };  
a();
```

Output?

- A) Hi
- B) Error
- C) delegate
- D) Nothing

 **Answer:** A

---

**32.**

```
Func<int,int,int> f = (a,b) => a+b;  
Console.WriteLine(f(2,3));
```

Output?

- A) 23
- B) 5
- C) Error
- D) 6

 **Answer:** B

---

**33.**

```
Action a = () => Console.WriteLine("A");  
a += () => Console.WriteLine("B");  
a();
```

Output?

- A) A
- B) B
- C) AB
- D) BA

 **Answer:** C

---

**34.**

```
Action a = () => Console.WriteLine("A");
a -= () => Console.WriteLine("A");
a();
```

Result?

- A) A
- B) Error
- C) Nothing
- D) Null

 **Answer:** A

 (Different instance, so not removed)

---

**35.**

```
Func<int> f = () => 10;
```

Return type?

- A) void
- B) int
- C) bool
- D) object

 **Answer:** B

---

**36.**

```
Func<int, bool> f = x => x > 0;
```

Which delegate?

- A) Action

- B) Func
- C) Predicate
- D) Event

 **Answer:** B

---

**37.**

```
Predicate<string> p = s => s.Length > 3;
```

Return type?

- A) int
- B) string
- C) bool
- D) void

 **Answer:** C

---

**38.**

```
Action<int,int> a = (x,y) => Console.WriteLine(x+y);
```

Parameters?

- A) 1
- B) 2
- C) 3
- D) Unlimited

 **Answer:** B

---

**39.**

```
Action a = null;  
a?.Invoke();
```

Result?

- A) Exception
- B) Compile error
- C) Safe, nothing happens
- D) Null

 **Answer:** C

---

**40.**

```
Action a = () => Console.WriteLine("X");  
a.Invoke();
```

Equivalent to?

- A) a();
- B) a.Call();
- C) Call(a);
- D) Error

 **Answer:** A

---

**41. Which supports multicast?**

- A) Func
- B) Action
- C) Custom delegate
- D) All delegates

 **Answer:** D

---

**42.**

```
Func<int,int> f = x => { return x*x; };
```

Valid?

- A) Yes
- B) No
- C) Warning
- D) Runtime error

 **Answer:** A

---

**43.**

```
Action<int> a = x => x++;
```

Compile?

- A) Yes
- B) No

 **Answer:** A

---

#### **44. Which is NOT true?**

- A) Func can return value
- B) Action returns void
- C) Predicate returns bool
- D) Action supports return

 **Answer:** D

---

#### **45.**

```
delegate int D(int x);
```

Signature requires?

- A) Same name
- B) Same return & parameters
- C) Same access modifier
- D) Same class

 **Answer:** B

---

#### **46. Which avoids custom delegate?**

- A) Func
- B) Action
- C) Predicate
- D) All

 **Answer:** D

---

#### **47. Which is faster to write?**

- A) Anonymous method
- B) Lambda
- C) Delegate class
- D) Interface

 **Answer:** B

---

**48.**

```
Func<int,int> f = delegate(int x){ return x+1; };
```

This is:

- A) Lambda
- B) Anonymous method
- C) Method group
- D) Error

 **Answer:** B

---

**49. Which improves readability most?**

- A) Anonymous methods
- B) Lambdas
- C) Custom delegates
- D) Events

 **Answer:** B

---

**50. Which is CLS compliant?**

- A) Delegates
- B) Lambdas
- C) Anonymous methods
- D) All

 **Answer:** D

---

## **HARD LEVEL (25 MCQs – TRICKY + SNIPPETS)**

**51.**

```
Action a = () => Console.WriteLine("A");
a += () => Console.WriteLine("B");
a -= () => Console.WriteLine("A");
a();
```

Output?

- A) A
- B) B

- C) AB
- D) Nothing

 **Answer:** C  
 (Removal fails – different instance)

---

**52.**

```
Func<int> f = () => 10;  
f += () => 20;  
Console.WriteLine(f());
```

Output?

- A) 10
- B) 20
- C) 1020
- D) Error

 **Answer:** B  
 (Last return value)

---

**53.**

```
Func<int, int> f = x => x++;  
Console.WriteLine(f(5));
```

Output?

- A) 5
- B) 6
- C) Error
- D) Undefined

 **Answer:** A

---

**54.**

```
Action<int> a = x => Console.WriteLine(x);  
Action<int> b = a;  
b(5);
```

Output?

- A) Error

- B) 5
- C) Nothing
- D) a

 **Answer:** B

---

### 55.

```
Action a = () => throw new Exception();  
a();
```

Result?

- A) Compile error
- B) Runtime exception
- C) Ignored
- D) Warning

 **Answer:** B

---

### 56. Which is resolved at runtime?

- A) Method group binding
- B) Delegate invocation
- C) Lambda syntax
- D) Generic inference

 **Answer:** B

---

### 57. Which returns only bool?

- A) Func<T,bool>
- B) Action<bool>
- C) Predicate<T>
- D) All

 **Answer:** C

---

### 58.

```
Func<int,int> f = x => x*2;  
Delegate d = f;
```

Valid?

- A) Yes
- B) No

 Answer: A

---

### 59. Which is immutable?

- A) Delegate reference
- B) Invocation list
- C) Multicast delegate
- D) Lambda body

 Answer: B

---

### 60.

```
Action a = null;  
a();
```

Result?

- A) Nothing
- B) Compile error
- C) Runtime exception
- D) Warning

 Answer: C

---

### 61. Which is safer?

- A) Direct method call
- B) Delegate with null check
- C) Reflection
- D) dynamic call

 Answer: B

---

### 62. Which supports event-driven programming?

- A) Interfaces
- B) Delegates

- A) Generics
- B) Tuples

 **Answer:** B

---

### **63. Which is best for callbacks?**

- A) Events
- B) Delegates
- C) Interfaces
- D) Classes

 **Answer:** B

---

### **64. Which avoids boilerplate code?**

- A) Anonymous methods
- B) Lambdas
- C) Delegates
- D) Interfaces

 **Answer:** B

---

### **65. Which is fastest execution?**

- A) Lambda
- B) Anonymous method
- C) Delegate
- D) Same (compiled to same IL)

 **Answer:** D

---

### **66. Which creates inline function?**

- A) Delegate
- B) Lambda
- C) Method
- D) Interface

 **Answer:** B

---

## **67. Which supports LINQ?**

- A) Delegates
- B) Lambdas
- C) Anonymous methods
- D) All

 **Answer:** D

---

## **68. Which is preferred in CCEE answers?**

- A) Syntax
- B) Concept clarity
- C) Advanced tricks
- D) UI examples

 **Answer:** B

---

## **69. Which breaks multicast?**

- A) Return values
- B) void return
- C) Action delegate
- D) Predicate

 **Answer:** A

---

## **70. Which is NOT true?**

- A) Action supports multicast
- B) Func supports multicast
- C) Return value of multicast Func is last one
- D) Predicate supports multicast safely

 **Answer:** D

---

## **71.**

```
Func<int,int> f = x => x*2;  
Console.WriteLine(f.Method.Name);
```

Prints?

- A) x
- B) Main
- C) Compiler generated name
- D) Error

 **Answer:** C

---

## 72. Which improves maintainability?

- A) Hardcoded logic
- B) Delegates
- C) Lambdas
- D) B + C

 **Answer:** D

---

## 73. Which reduces code size?

- A) Interfaces
- B) Delegates
- C) Lambdas
- D) Classes

 **Answer:** C

---

## 74. Which is MOST asked in CCEE?

- A) Events
- B) Multicast delegates output
- C) Unsafe code
- D) Reflection

 **Answer:** B

---

## 75. Which is BEST practice?

- A) Avoid delegates
- B) Use lambdas with Action/Func
- C) Always write custom delegates
- D) Use anonymous methods only

 **Answer:** B

# SESSION 9 – Exception Handling & Events (CCEE MCQs)

---

## EASY LEVEL (25 MCQs)

### 1. Exceptions are used to handle:

- A) Compile-time errors
- B) Runtime errors
- C) Syntax errors
- D) Logical errors only

 Answer: B

---

### 2. Which block always executes?

- A) try
- B) catch
- C) finally
- D) throw

 Answer: C

---

### 3. Which keyword is used to explicitly raise an exception?

- A) new
- B) raise
- C) throw
- D) error

 Answer: C

---

### 4. Which namespace contains Exception class?

- A) System.IO
- B) System.Text

- C) System
- D) System.Collections

 **Answer:** C

---

## 5. Which block handles exceptions?

- A) try
- B) catch
- C) finally
- D) throw

 **Answer:** B

---

## 6. Which is a checked exception in C#?

- A) NullReferenceException
- B) IndexOutOfRangeException
- C) FileNotFoundException
- D) None (C# has no checked exceptions)

 **Answer:** D

---

## 7. Which is TRUE about C# exceptions?

- A) Checked exceptions are mandatory
- B) Unchecked exceptions only
- C) No checked exceptions
- D) Both checked & unchecked

 **Answer:** C

---

## 8. User-defined exception should inherit from:

- A) object
- B) ApplicationException
- C) Exception
- D) SystemError

 **Answer:** C

---

## **9. Which keyword is used to create custom exception?**

- A) exception
- B) throw
- C) class
- D) new

 **Answer:** C

---

## **10. Events are based on:**

- A) Interfaces
- B) Delegates
- C) Classes
- D) Structs

 **Answer:** B

---

## **11. Which keyword declares an event?**

- A) delegate
- B) handler
- C) event
- D) notify

 **Answer:** C

---

## **12. Which pattern is commonly used for events?**

- A) MVC
- B) Observer
- C) Singleton
- D) Factory

 **Answer:** B

---

## **13. Which delegate is commonly used for events?**

- A) Action
- B) Func
- C) Predicate
- D) EventHandler

 Answer: D

---

#### **14. Which method raises an event?**

- A) Raise()
- B) Invoke()
- C) Call()
- D) Fire()

 Answer: B

---

#### **15. Which block is optional?**

- A) try
- B) catch
- C) finally
- D) try & catch both

 Answer: C

---

#### **16. Which is a bad practice?**

- A) Catch specific exceptions
- B) Use finally for cleanup
- C) Catch System.Exception always
- D) Throw meaningful exceptions

 Answer: C

---

#### **17. Which ensures resource cleanup?**

- A) catch
- B) throw
- C) finally
- D) event

 Answer: C

---

#### **18. Which event keyword prevents external invocation?**

- A) private
- B) protected
- C) sealed
- D) event

 **Answer:** D

---

### **19. Which event argument class is base?**

- A) EventArgs
- B) Object
- C) Exception
- D) Delegate

 **Answer:** A

---

### **20. Which is TRUE?**

- A) finally runs only if exception occurs
- B) finally runs only if no exception
- C) finally always runs
- D) finally is optional and unsafe

 **Answer:** C

---

### **21. Which exception occurs when dividing by zero?**

- A) ArithmeticException
- B) DivideByZeroException
- C) InvalidOperationException
- D) MathException

 **Answer:** B

---

### **22. Which statement rethrows same exception?**

- A) throw ex;
- B) throw;
- C) new Exception();
- D) catch;

 **Answer:** B

---

## 23. Which is NOT an exception handling keyword?

- A) try
- B) catch
- C) finally
- D) check

 Answer: D

---

## 24. Which supports multiple handlers?

- A) try block
- B) catch blocks
- C) finally block
- D) throw block

 Answer: B

---

## 25. Which improves program reliability?

- A) Ignoring exceptions
- B) Using exception handling
- C) Using return codes
- D) Avoiding errors

 Answer: B

---

## MEDIUM LEVEL (25 MCQs – MANY SNIPPETS)

### 26.

```
try {  
    int x = 10/0;  
}  
catch {  
    Console.WriteLine("E");  
}
```

Output?  
A) 0

- B) E
- C) Exception
- D) Nothing

 **Answer:** B

---

**27.**

```
try {
    Console.WriteLine("A");
}
finally {
    Console.WriteLine("B");
}
```

Output?

- A) A
- B) B
- C) AB
- D) BA

 **Answer:** C

---

**28.**

```
try {
    throw new Exception();
}
catch(Exception e) {
    Console.WriteLine("C");
}
```

Output?

- A) Exception
- B) C
- C) Runtime error
- D) Nothing

 **Answer:** B

---

**29.**

```
try {
    Console.WriteLine("T");
}
catch {
    Console.WriteLine("C");
}
finally {
    Console.WriteLine("F");
}
```

Output?

- A) TF
- B) TCF
- C) CF
- D) T

 **Answer:** A

---

**30.**

```
throw new NullReferenceException();
```

This is:

- A) Compile error
- B) Runtime exception
- C) Syntax error
- D) Warning

 **Answer:** B

---

**31.**

```
catch(Exception ex) {
    throw;
}
```

Effect?

- A) Loses stack trace
- B) Preserves stack trace
- C) New exception
- D) Compile error

 **Answer:** B

---

**32.**

```
catch(Exception ex) {  
    throw ex;  
}
```

Effect?

- A) Preserves stack trace
- B) Loses stack trace
- C) No difference
- D) Compile error

 **Answer:** B

---

**33.**

```
class MyEx : Exception {}
```

Valid?

- A) Yes
- B) No
- C) Warning
- D) Runtime error

 **Answer:** A

---

**34.**

```
event Action E;
```

Meaning?

- A) Method
- B) Delegate
- C) Event
- D) Property

 **Answer:** C

---

**35.**

```
E?.Invoke();
```

Purpose?

- A) Raise event safely
- B) Catch exception
- C) Dispose event
- D) Clear handlers

 **Answer:** A

---

**36.**

```
try { }  
finally { }
```

Valid?

- A) Yes
- B) No
- C) Warning
- D) Runtime error

 **Answer:** A

---

**37.**

```
try { }  
catch { }
```

Valid?

- A) Yes
- B) No
- C) Warning
- D) Runtime error

 **Answer:** A

---

**38.**

```
try { }
```

Valid?

- A) Yes
- B) No

- C) Warning
- D) Runtime error

 **Answer:** B

---

**39.**

```
event EventHandler MyEvent;
```

Handler signature?

- A) void()
- B) void(object,EventArgs)
- C) bool()
- D) int()

 **Answer:** B

---

**40.**

```
MyEvent += Handler;
```

This is:

- A) Raising event
- B) Subscribing
- C) Unsubscribing
- D) Overriding

 **Answer:** B

---

**41.**

```
MyEvent -= Handler;
```

This is:

- A) Raising
- B) Subscribing
- C) Unsubscribing
- D) Hiding

 **Answer:** C

---

## 42. Which block runs even after return?

- A) try
- B) catch
- C) finally
- D) throw

 Answer: C

---

## 43. Which is best practice?

- A) Empty catch block
- B) Catch specific exceptions
- C) Catch all exceptions
- D) Ignore exceptions

 Answer: B

---

## 44.

```
try {
    return;
}
finally {
    Console.WriteLine("F");
}
```

Output?

- A) Nothing
- B) F
- C) Error
- D) return

 Answer: B

---

## 45. Which is TRUE?

- A) finally may not execute
- B) finally executes always
- C) finally executes only on exception
- D) finally is optional and unsafe

 Answer: B

---

## **46. Which raises event internally?**

- A) Subscriber
- B) Publisher
- C) CLR
- D) OS

 **Answer:** B

---

## **47. Which avoids NullReferenceException for events?**

- A) if(event!=null)
- B) try-catch
- C) ?.Invoke()
- D) All

 **Answer:** D

---

## **48. Which should NOT be thrown?**

- A) Custom exception
- B) Exception
- C) System.Exception directly
- D) InvalidOperationException

 **Answer:** C

---

## **49. Which is CLS compliant?**

- A) Events
- B) Custom exceptions
- C) try-catch
- D) All

 **Answer:** D

---

## **50. Which supports multiple subscribers?**

- A) Event
- B) Method

- C) Property
- D) Field

 **Answer:** A

---

## **HARD LEVEL (25 MCQs – TRICKY + SNIPPETS)**

**51.**

```
try {  
    throw new Exception("X");  
}  
catch {  
    throw;  
}
```

Stack trace?

- A) Lost
- B) Preserved
- C) New
- D) Empty

 **Answer:** B

---

**52.**

```
try {  
    throw new Exception();  
}  
catch(Exception ex) {  
    throw ex;  
}
```

Stack trace?

- A) Preserved
- B) Lost
- C) Same
- D) None

 **Answer:** B

---

**53.**

```
Action a = () => throw new Exception();
try {
    a();
}
catch {
    Console.WriteLine("C");
}
```

Output?

- A) Exception
- B) C
- C) Nothing
- D) Runtime crash

 **Answer:** B

---

**54.**

```
event Action E;
E();
```

Result?

- A) Safe
- B) Compile error
- C) Runtime exception
- D) Warning

 **Answer:** C

---

**55.**

```
event Action E;
E?.Invoke();
```

Result?

- A) Safe
- B) Error
- C) Exception
- D) Warning

 **Answer:** A

---

## 56. Which is NOT recommended?

- A) Throw specific exceptions
- B) Catch and rethrow using throw
- C) Catch and ignore
- D) Use finally

 Answer: C

---

## 57. Which is executed first?

- A) finally
- B) catch
- C) try
- D) throw

 Answer: C

---

## 58.

```
try {  
    throw new Exception();  
}  
finally {  
    Console.WriteLine("F");  
}
```

Output?

- A) Exception
- B) F
- C) Nothing
- D) Runtime error

 Answer: B

---

## 59. Which pattern do events follow?

- A) MVC
- B) Observer
- C) Singleton
- D) Adapter

 **Answer:** B

---

**60.**

```
event EventHandler E;  
E(this,EventArgs.Empty);
```

Allowed?

- A) Always
- B) Only inside class
- C) Only outside
- D) Never

 **Answer:** B

---

**61. Which improves debuggability?**

- A) throw ex
- B) throw
- C) ignore exception
- D) catch all

 **Answer:** B

---

**62. Which should be sealed?**

- A) Exception class
- B) Custom exception class
- C) Event handler
- D) finally block

 **Answer:** B

---

**63. Which causes app crash?**

- A) Unhandled exception
- B) Handled exception
- C) finally block
- D) catch block

 **Answer:** A

---

#### **64. Which is slow?**

- A) try-catch without exception
- B) try-catch with exception
- C) finally
- D) throw keyword

 **Answer:** B

---

#### **65.**

`catch(Exception) { }`

This is:

- A) Good practice
- B) Bad practice
- C) Mandatory
- D) Required

 **Answer:** B

---

#### **66. Which prevents event misuse?**

- A) public delegate
- B) event keyword
- C) Action delegate
- D) Func delegate

 **Answer:** B

---

#### **67. Which ensures cleanup?**

- A) catch
- B) finally
- C) event
- D) delegate

 **Answer:** B

---

#### **68. Which allows multiple handlers?**

- A) Method
- B) Delegate
- C) Event
- D) Property

 **Answer:** C

---

## 69. Which improves maintainability?

- A) Proper exception hierarchy
- B) Empty catch blocks
- C) Ignoring errors
- D) Using return codes

 **Answer:** A

---

## 70. Which is BEST practice?

- A) Handle all exceptions silently
- B) Log and rethrow if needed
- C) Ignore finally
- D) Use throw ex

 **Answer:** B

---

## 71.

```
try {
    throw new Exception();
}
catch {
}
Console.WriteLine("A");
```

Output?

- A) A
- B) Exception
- C) Nothing
- D) Runtime error

 **Answer:** A

---

## **72. Which is NOT allowed?**

- A) Multiple catch blocks
- B) finally without catch
- C) try without catch/finally
- D) Nested try

 **Answer:** C

---

## **73. Which supports event-driven architecture?**

- A) Classes
- B) Delegates & events
- C) Structs
- D) Enums

 **Answer:** B

---

## **74. Which is MOST asked in CCEE?**

- A) File IO exceptions
- B) try-catch-finally flow
- C) Custom exception syntax
- D) UI events

 **Answer:** B

---

## **75. Which is KEY for CCEE scoring?**

- A) Syntax memorization
- B) Output prediction
- C) UI design
- D) Framework version

 **Answer:** B

 **SESSION 10 – LINQ, Anonymous Types, Extensions, PLINQ (CCEE MCQs)**

---

## EASY LEVEL (25 MCQs)

### 1. Anonymous types are created using:

- A) class
- B) struct
- C) var
- D) new with name

 Answer: C

---

### 2. Anonymous types are:

- A) Mutable
- B) Immutable
- C) Static
- D) Inheritable

 Answer: B

---

### 3. Extension methods must be:

- A) Instance methods
- B) Static methods
- C) Virtual methods
- D) Abstract methods

 Answer: B

---

### 4. Extension methods are defined inside:

- A) Class
- B) Interface
- C) Static class
- D) Struct

 Answer: C

---

### 5. Which keyword is used for partial classes?

- A) sealed
- B) split
- C) partial
- D) static

 Answer: C

---

## 6. Partial classes must be:

- A) In same file
- B) In different files only
- C) In same assembly
- D) In same namespace only

 Answer: C

---

## 7. Partial methods must be:

- A) public
- B) private
- C) virtual
- D) static

 Answer: B

---

## 8. LINQ stands for:

- A) Language Integrated Query
- B) Logical Integrated Query
- C) Local Interface Query
- D) Language Internal Query

 Answer: A

---

## 9. LINQ to Objects works on:

- A) Database
- B) XML
- C) In-memory collections
- D) Files

 Answer: C

---

## **10. LINQ queries return:**

- A) Values immediately
- B) Objects
- C) IEnumerable
- D) Arrays only

 **Answer:** C

---

## **11. Which keyword introduces a LINQ query?**

- A) select
- B) from
- C) where
- D) orderby

 **Answer:** B

---

## **12. Deferred execution means:**

- A) Query runs immediately
- B) Query runs on declaration
- C) Query runs on iteration
- D) Query runs at compile time

 **Answer:** C

---

## **13. Which LINQ method executes immediately?**

- A) Where
- B) Select
- C) ToList
- D) OrderBy

 **Answer:** C

---

## **14. Which LINQ method filters data?**

- A) Select
- B) Where

- A) OrderBy
- B) GroupBy

 **Answer:** B

---

### **15. Which LINQ method projects data?**

- A) Where
- B) Select
- C) GroupBy
- D) Join

 **Answer:** B

---

### **16. PLINQ stands for:**

- A) Parallel LINQ
- B) Partial LINQ
- C) Protected LINQ
- D) Programmatic LINQ

 **Answer:** A

---

### **17. PLINQ is used for:**

- A) UI queries
- B) Parallel execution
- C) Database access
- D) File handling

 **Answer:** B

---

### **18. Which operator enables parallel LINQ?**

- A) AsQueryable()
- B) AsParallel()
- C) Parallel()
- D) RunParallel()

 **Answer:** B

---

## **19. LINQ queries are:**

- A) Platform dependent
- B) Language dependent
- C) Type safe
- D) Dynamic

 **Answer:** C

---

## **20. Which requires System.Linq namespace?**

- A) foreach
- B) LINQ methods
- C) Arrays
- D) Lists

 **Answer:** B

---

## **21. Which is NOT a LINQ method?**

- A) Select
- B) Where
- C) Filter
- D) OrderBy

 **Answer:** C

---

## **22. Which is true?**

- A) LINQ modifies source collection
- B) LINQ returns new sequence
- C) LINQ deletes data
- D) LINQ sorts permanently

 **Answer:** B

---

## **23. Which supports query syntax?**

- A) Arrays
- B) IEnumerable
- C) Delegates
- D) Events

 **Answer:** B

---

## **24. Which improves readability?**

- A) Extension methods
- B) LINQ
- C) Anonymous types
- D) All

 **Answer:** D

---

## **25. Which is MOST used in CCEE?**

- A) PLINQ
- B) Deferred execution
- C) LINQ output MCQs
- D) Partial methods

 **Answer:** C

---

## **MEDIUM LEVEL (25 MCQs – MANY SNIPPETS)**

### **26.**

```
var x = new { Id = 1, Name = "A" };
```

Type of x?

- A) class
- B) object
- C) Anonymous type
- D) dynamic

 **Answer:** C

---

### **27.**

```
static class Ex {
    public static int Double(this int x) => x*2;
}
Console.WriteLine(5.Double());
```

Output?

- A) 5
- B) 10
- C) Error
- D) 0

 Answer: B

---

**28.**

```
partial class A {  
    void M(){}
}  
partial class A {  
    void N(){}
}
```

Result?

- A) Error
- B) Two classes
- C) One combined class
- D) Warning

 Answer: C

---

**29.**

```
partial void P();
```

Where allowed?

- A) Interface
- B) Abstract class
- C) Partial class
- D) Static class

 Answer: C

---

**30.**

```
int[] a = {1,2,3,4};  
var q = from x in a where x>2 select x;
```

When does q execute?

- A) Declaration
- B) Compilation
- C) Iteration
- D) Assignment

 **Answer:** C

---

**31.**

```
var q = a.Where(x => x>2);
```

Execution?

- A) Immediate
- B) Deferred
- C) Compile time
- D) Static

 **Answer:** B

---

**32.**

```
var q = a.Where(x=>x>2).ToList();
```

Execution?

- A) Deferred
- B) Immediate
- C) Lazy
- D) Parallel

 **Answer:** B

---

**33.**

```
var q = from x in a select x*2;
```

Result type?

- A) int[]
- B) IEnumerable<int>
- C) List<int>
- D) object

 **Answer:** B

---

**34.**

```
var q = a.Select(x => new { V = x });
```

Creates?

- A) class
- B) struct
- C) Anonymous type
- D) Tuple

 **Answer:** C

---

**35.**

```
var q = a.OrderBy(x => x);
```

Sorting?

- A) Descending
- B) Ascending
- C) Random
- D) Permanent

 **Answer:** B

---

**36.**

```
var q = a.GroupBy(x => x%2);
```

Groups by?

- A) Value
- B) Index
- C) Even/Odd
- D) Size

 **Answer:** C

---

**37.**

```
var q = a.Take(2);
```

Result?

- A) First 2 elements
- B) Last 2 elements
- C) Random
- D) Error

 Answer: A

---

38.

```
var q = a.Skip(2);
```

Result?

- A) First 2 skipped
- B) Last 2 skipped
- C) Error
- D) Sorted

 Answer: A

---

39.

```
var q = a.Any(x => x>3);
```

Return type?

- A) int
- B) bool
- C) IEnumerable
- D) object

 Answer: B

---

40.

```
var q = a.Count();
```

Execution?

- A) Deferred
- B) Immediate
- C) Parallel
- D) Lazy

 Answer: B

---

**41.**

```
var q = a.AsParallel().Where(x=>x>2);
```

This is?

- A) LINQ
- B) PLINQ
- C) Deferred only
- D) Sequential

 **Answer:** B

---

**42.**

```
var q = a.First();
```

If empty?

- A) null
- B) 0
- C) Exception
- D) false

 **Answer:** C

---

**43.**

```
var q = a.FirstOrDefault();
```

If empty?

- A) Exception
- B) Default value
- C) null only
- D) Error

 **Answer:** B

---

**44.**

```
var q = a.Single();
```

If more than one?

- A) First returned
- B) Exception
- C) null
- D) Last returned

 **Answer:** B

---

#### **45. Which is immediate execution?**

- A) Where
- B) Select
- C) ToArray
- D) OrderBy

 **Answer:** C

---

#### **46. Which uses deferred execution?**

- A) Count
- B) Sum
- C) Where
- D) ToList

 **Answer:** C

---

#### **47. Which allows parallel execution?**

- A) LINQ
- B) PLINQ
- C) Query syntax
- D) Deferred execution

 **Answer:** B

---

#### **48.**

```
var q = a.Select((x,i)=>x+i);
```

i is?

- A) Value
- B) Index

- C) Key
- D) Counter

 **Answer:** B

---

#### **49. Which is immutable?**

- A) Anonymous type
- B) List
- C) Array
- D) Dictionary

 **Answer:** A

---

#### **50. Which improves performance?**

- A) Deferred execution
- B) PLINQ on small data
- C) Immediate execution
- D) Boxing

 **Answer:** A

---

### **HARD LEVEL (25 MCQs – TRICKY + SNIPPETS)**

#### **51.**

```
int[] a = {1,2,3};  
var q = a.Where(x=>x>1);  
a[1] = 10;  
foreach(var x in q) Console.WriteLine(x);
```

Output?

- A) 23
- B) 210
- C) 103
- D) 101

 **Answer:** B  
 (Deferred execution)

---

**52.**

```
var q = a.Where(x=>x>1).ToList();
a[1]=10;
```

q contains?

- A) Updated values
- B) Original values
- C) Error
- D) Empty

 **Answer:** B

---

**53.**

```
var q = from x in a select x;
```

This is:

- A) Immediate
- B) Deferred
- C) Compiled
- D) Cached

 **Answer:** B

---

**54.**

```
var q = a.AsParallel().OrderBy(x=>x);
```

Order guaranteed?

- A) Yes
- B) No
- C) Random
- D) Compile error

 **Answer:** B

---

**55.**

```
var q = a.AsParallel().AsOrdered();
```

Purpose?

- A) Faster
- B) Preserve order
- C) Deferred
- D) Sequential

 **Answer:** B

---

## 56. Which is FALSE?

- A) Anonymous types are immutable
- B) Extension methods need static class
- C) LINQ always executes immediately
- D) PLINQ uses multiple threads

 **Answer:** C

---

## 57.

```
partial void M();
```

If not implemented?

- A) Runtime error
- B) Compile error
- C) Removed at compile time
- D) Warning

 **Answer:** C

---

## 58.

```
var q = a.Select(x=>x/0);
```

When exception?

- A) Declaration
- B) Compilation
- C) Iteration
- D) Assignment

 **Answer:** C

---

## 59.

```
var q = a.Where(x=>x>2);  
Console.WriteLine(q);
```

Output?

- A) Values
- B) Type name
- C) Error
- D) null

 **Answer:** B

---

**60.**

```
var q = a.AsParallel().ForAll(Console.WriteLine);
```

Execution?

- A) Sequential
- B) Parallel
- C) Deferred
- D) Ordered

 **Answer:** B

---

**61. Which causes side effects?**

- A) Deferred execution
- B) Immediate execution
- C) Pure functions
- D) Immutability

 **Answer:** A

---

**62. Which is best practice?**

- A) Modify source in LINQ
- B) Use pure expressions
- C) Use PLINQ everywhere
- D) Ignore execution model

 **Answer:** B

---

### **63. Which returns scalar?**

- A) Select
- B) Where
- C) Count
- D) OrderBy

 **Answer:** C

---

### **64.**

```
var q = a.Select(x=>x).Select(x=>x);
```

Execution count?

- A) Once
- B) Twice
- C) Per iteration
- D) Error

 **Answer:** C

---

### **65. Which is thread-safe?**

- A) List
- B) LINQ
- C) PLINQ queries
- D) ArrayList

 **Answer:** C

---

### **66. Which is evaluated lazily?**

- A) Sum
- B) Max
- C) Where
- D) ToList

 **Answer:** C

---

### **67. Which improves abstraction?**

- A) Anonymous types
- B) LINQ
- C) Extension methods
- D) All

 Answer: D

---

## 68. Which is fastest for small collections?

- A) PLINQ
- B) LINQ
- C) Manual loop
- D) Reflection

 Answer: C

---

## 69. Which breaks deferred execution?

- A) foreach
- B) ToList
- C) Where
- D) Select

 Answer: B

---

## 70. Which is MOST asked in CCEE?

- A) PLINQ syntax
- B) Deferred execution output
- C) Partial methods
- D) Anonymous types syntax

 Answer: B

---

## 71.

```
var q = a.Where(x=>x>1);
Console.WriteLine(q.Count());
Console.WriteLine(q.Count());
```

Iterations?  
A) Once

- A) Twice
- C) Cached
- D) Error

 **Answer:** B

---

## 72. Which improves code reuse?

- A) Partial classes
- B) Extension methods
- C) LINQ
- D) All

 **Answer:** D

---

## 73. Which is compile-time feature?

- A) Deferred execution
- B) Partial methods
- C) PLINQ
- D) LINQ queries

 **Answer:** B

---

## 74. Which should be avoided?

- A) Heavy logic in LINQ
- B) Using ToList wisely
- C) Using FirstOrDefault
- D) Using Select

 **Answer:** A

---

## 75. Which is KEY for CCEE scoring?

- A) Memorizing syntax
- B) Understanding execution model
- C) Writing full programs
- D) UI coding

 **Answer:** B

# SESSION 11 – Assemblies, Reflection, Attributes, File I/O (CCEE MCQs)

---

## EASY LEVEL (20 MCQs)

### 1. A shared assembly is also known as:

- A) Private assembly
- B) Global assembly
- C) Strong-named assembly
- D) Local assembly

 Answer: C

---

### 2. Shared assemblies are usually stored in:

- A) Bin folder
- B) Application folder
- C) GAC
- D) Temp folder

 Answer: C

---

### 3. GAC stands for:

- A) Global Application Cache
- B) Global Assembly Cache
- C) Generic Assembly Container
- D) Global Access Container

 Answer: B

---

### 4. Custom attributes are used to:

- A) Store data
- B) Add metadata

- A) Handle exceptions
- B) Improve performance

 **Answer:** B

---

## **5. Which namespace is used for attributes?**

- A) System.IO
- B) System.Reflection
- C) System
- D) System.ComponentModel

 **Answer:** C

---

## **6. Reflection is used to:**

- A) Compile code
- B) Execute SQL
- C) Inspect metadata at runtime
- D) Optimize code

 **Answer:** C

---

## **7. Which namespace supports reflection?**

- A) System.IO
- B) System.Reflection
- C) System.Linq
- D) System.Collections

 **Answer:** B

---

## **8. Which class represents an assembly?**

- A) Module
- B) Type
- C) Assembly
- D) Attribute

 **Answer:** C

---

## **9. Reflection works at:**

- A) Compile time
- B) Link time
- C) Runtime
- D) Design time

 **Answer:** C

---

## **10. Which method loads an assembly dynamically?**

- A) Assembly.GetTypes()
- B) Assembly.Load()
- C) Assembly.Create()
- D) Assembly.Open()

 **Answer:** B

---

## **11. File handling classes are in:**

- A) System
- B) System.Collections
- C) System.IO
- D) System.Text

 **Answer:** C

---

## **12. Which class represents a file?**

- A) FileInfo
- B) DirectoryInfo
- C) DriveInfo
- D) Stream

 **Answer:** A

---

## **13. Which class represents a directory?**

- A) File
- B) Directory
- C) DirectoryInfo
- D) Path

 Answer: C

---

**14. Which stream is used for reading text?**

- A) BinaryWriter
- B) StreamWriter
- C) StreamReader
- D) FileStream

 Answer: C

---

**15. Which stream is used for writing text?**

- A) StreamReader
- B) StreamWriter
- C) BinaryReader
- D) FileReader

 Answer: B

---

**16. Which class provides static file operations?**

- A) FileInfo
- B) File
- C) DirectoryInfo
- D) Stream

 Answer: B

---

**17. Which class represents logical drives?**

- A) Directory
- B) DriveInfo
- C) File
- D) Path

 Answer: B

---

**18. Which attribute specifies attribute usage rules?**

- A) Serializable
- B) AttributeUsage
- C) Obsolete
- D) Conditional

 Answer: B

---

## 19. Which attribute marks deprecated code?

- A) Serializable
- B) Obsolete
- C) CLSCompliant
- D) Flags

 Answer: B

---

## 20. File streams work with:

- A) Text only
- B) Binary data
- C) Metadata
- D) Assemblies

 Answer: B

---

## MEDIUM LEVEL (20 MCQs – WITH SNIPPETS)

### 21.

```
[Obsolete]  
class A {}
```

Meaning?

- A) Class is deleted
- B) Class is outdated
- C) Compile error
- D) Runtime error

 Answer: B

---

### 22.

```
class MyAttr : Attribute {}
```

This defines:

- A) Exception
- B) Custom attribute
- C) Delegate
- D) Interface

 **Answer:** B

---

**23.**

```
var asm = Assembly.GetExecutingAssembly();
```

Returns?

- A) Loaded DLL
- B) Current assembly
- C) Referenced assembly
- D) GAC assembly

 **Answer:** B

---

**24.**

```
Type t = typeof(string);
```

t represents?

- A) Object
- B) Class
- C) Metadata of string
- D) Value

 **Answer:** C

---

**25.**

```
Assembly.Load("MyLib");
```

This does?

- A) Compiles assembly
- B) Loads assembly at runtime

- C) Creates object
- D) Reads metadata only

 **Answer:** B

---

**26.**

`Attribute.GetCustomAttributes(t);`

Returns?

- A) Methods
- B) Fields
- C) Applied attributes
- D) Properties

 **Answer:** C

---

**27.**

`File.Exists("a.txt");`

Returns?

- A) File object
- B) Boolean
- C) Stream
- D) Exception

 **Answer:** B

---

**28.**

`File.ReadAllText("a.txt");`

Returns?

- A) byte[]
- B) string
- C) char[]
- D) Stream

 **Answer:** B

---

**29.**

```
File.WriteAllText("a.txt", "Hi");
```

Does?

- A) Appends text
- B) Overwrites file
- C) Reads file
- D) Deletes file

 **Answer:** B

---

**30.**

```
Directory.Exists("Test");
```

Checks?

- A) File existence
- B) Drive existence
- C) Directory existence
- D) Stream state

 **Answer:** C

---

**31.**

```
using(var sw = new StreamWriter("a.txt")) {}
```

Purpose of using?

- A) Faster IO
- B) Automatic Dispose
- C) Thread safety
- D) Buffering

 **Answer:** B

---

**32.**

```
FileStream fs = new FileStream("a.txt", FileMode.Open);
```

If file missing?

- A) New file created
- B) Exception

- C) null
- D) Empty stream

 **Answer:** B

---

### **33. Which FileMode creates file if not exists?**

- A) Open
- B) Append
- C) Create
- D) Truncate

 **Answer:** C

---

### **34. Which reflection class describes methods?**

- A) Type
- B) MethodInfo
- C) PropertyInfo
- D) FieldInfo

 **Answer:** B

---

### **35.**

```
foreach(var t in asm.GetTypes())
```

Iterates over?

- A) Objects
- B) Classes in assembly
- C) Files
- D) Streams

 **Answer:** B

---

### **36. Which is TRUE?**

- A) Reflection is compile-time
- B) Reflection is slow
- C) Reflection improves performance
- D) Reflection avoids metadata

 **Answer:** B

---

### **37. Which attribute controls multiple usage?**

- A) Obsolete
- B) Flags
- C) AttributeUsage
- D) Serializable

 **Answer:** C

---

### **38.**

`[AttributeUsage(AttributeTargets.Class)]`

Applies to?

- A) Methods
- B) Fields
- C) Classes
- D) Assemblies

 **Answer:** C

---

### **39. Which class reads binary data?**

- A) StreamReader
- B) BinaryReader
- C) StreamWriter
- D) TextReader

 **Answer:** B

---

### **40. Which supports appending text?**

- A) FileMode.Create
- B) FileMode.Append
- C) FileMode.Open
- D) FileMode.Truncate

 **Answer:** B

---

## HARD LEVEL (20 MCQs – TRICKY + SNIPPETS)

41.

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple=true)]
```

Effect?

- A) Applied once
- B) Applied multiple times
- C) Runtime error
- D) Ignored

 Answer: B

---

42.

```
Assembly.LoadFrom("a.dll");
```

Loads from?

- A) GAC
- B) Local path
- C) Memory
- D) Network only

 Answer: B

---

43.

```
Type t = asm.GetType("MyClass");
```

Returns?

- A) Object
- B) Metadata
- C) Instance
- D) Stream

 Answer: B

---

44.

```
Activator.CreateInstance(t);
```

Creates?

- A) Metadata
- B) Object instance
- C) Assembly
- D) Stream

 **Answer:** B

---

#### **45. Which is NOT reflection usage?**

- A) Discover methods
- B) Dynamic loading
- C) Compile-time checking
- D) Invoke methods dynamically

 **Answer:** C

---

#### **46. Which is safest for file IO?**

- A) Manual Close
- B) using statement
- C) GC cleanup
- D) Destructor

 **Answer:** B

---

#### **47.**

`File.ReadAllText("a.txt");`

Returns?

- A) string
- B) string[]
- C) byte[]
- D) Stream

 **Answer:** B

---

#### **48.**

`Directory.GetFiles(".");`

Returns?

- A) File objects
- B) FileInfo[]
- C) string[]
- D) Streams

 Answer: C

---

#### 49. Which can cause security risk?

- A) File IO
- B) Reflection
- C) Attributes
- D) GAC

 Answer: B

---

#### 50. Which is FALSE?

- A) Shared assemblies require strong name
- B) Reflection uses metadata
- C) FileStream supports text only
- D) Custom attributes inherit Attribute

 Answer: C

---

#### 51.

```
using System.Reflection;
```

Required for?

- A) File IO
- B) LINQ
- C) Reflection
- D) Serialization

 Answer: C

---

#### 52. Which improves versioning?

- A) Private assembly
- B) Shared assembly

- C) Static assembly
- D) Dynamic assembly

 **Answer:** B

---

### **53. Which is slower?**

- A) Direct method call
- B) Reflection call
- C) Static call
- D) Inline call

 **Answer:** B

---

### **54.**

`FileMode.Truncate`

Does?

- A) Deletes file
- B) Clears file content
- C) Appends data
- D) Opens file

 **Answer:** B

---

### **55. Which should be avoided in performance-critical code?**

- A) LINQ
- B) Reflection
- C) Arrays
- D) Structs

 **Answer:** B

---

### **56. Which attribute affects compilation?**

- A) Obsolete
- B) Serializable
- C) CLSCompliant
- D) Conditional

 Answer: D

---

**57.**

```
[Serializable]  
class A {}
```

Purpose?

- A) Reflection
- B) Persistence
- C) Versioning
- D) Security

 Answer: B

---

**58. Which returns full path info?**

- A) File
- B) Path
- C) Stream
- D) DriveInfo

 Answer: B

---

**59. Which IO class is abstract?**

- A) FileStream
- B) Stream
- C) BinaryReader
- D) StreamReader

 Answer: B

---

**60. Which is MOST asked in CCEE?**

- A) Strong name syntax
- B) Reflection basics
- C) Advanced IO buffering
- D) Custom attribute internals

 Answer: B

# SESSION 12 – Threading, Tasks, Async/Await, TPL (CCEE MCQs)

---

## EASY LEVEL (20 MCQs)

### 1. Threading is used to:

- A) Improve UI design
- B) Execute multiple tasks concurrently
- C) Reduce memory
- D) Compile faster

 Answer: B

---

### 2. Which namespace contains Thread class?

- A) System.Threading
- B) System.Tasks
- C) System.Runtime
- D) System.Concurrent

 Answer: A

---

### 3. Which delegate is used to start a thread without parameters?

- A) Action
- B) Func
- C) ThreadStart
- D) ParameterizedThreadStart

 Answer: C

---

### 4. Which delegate allows passing parameter to a thread?

- A) ThreadStart
- B) Action

- C) Func
- D) ParameterizedThreadStart

 **Answer:** D

---

## **5. Which method starts a thread?**

- A) Run()
- B) Execute()
- C) Start()
- D) Begin()

 **Answer:** C

---

## **6. ThreadPool threads are:**

- A) Foreground threads
- B) Background threads
- C) UI threads
- D) Static threads

 **Answer:** B

---

## **7. Which keyword is used for synchronization?**

- A) sync
- B) lock
- C) wait
- D) mutex

 **Answer:** B

---

## **8. Which class provides low-level synchronization?**

- A) Thread
- B) Task
- C) Monitor
- D) ThreadPool

 **Answer:** C

---

## **9. Which class performs atomic operations?**

- A) Task
- B) Monitor
- C) Interlocked
- D) Thread

 **Answer:** C

---

## **10. Which namespace contains Task?**

- A) System.Threading
- B) System.Threading.Tasks
- C) System.Runtime
- D) System.Linq

 **Answer:** B

---

## **11. Task is:**

- A) OS-level thread
- B) Logical unit of work
- C) UI thread
- D) Process

 **Answer:** B

---

## **12. Which keyword marks async method?**

- A) await
- B) async
- C) task
- D) parallel

 **Answer:** B

---

## **13. Which keyword pauses execution asynchronously?**

- A) wait
- B) sleep
- C) await
- D) delay

 Answer: C

---

**14. async methods return:**

- A) void only
- B) Task / Task<T>
- C) Thread
- D) object

 Answer: B

---

**15. Which runs code in parallel easily?**

- A) Thread
- B) Task
- C) TPL
- D) Monitor

 Answer: C

---

**16. Which is recommended for new code?**

- A) Thread
- B) ThreadPool
- C) Task-based async
- D) Timer

 Answer: C

---

**17. Which is NOT thread-safe by default?**

- A) lock
- B) Interlocked
- C) List<T>
- D) Monitor

 Answer: C

---

**18. Which method blocks current thread?**

- A) await
- B) Task.Delay
- C) Thread.Sleep
- D) async

 **Answer:** C

---

### **19. Which allows returning value from background work?**

- A) Thread
- B) Task
- C) Monitor
- D) ThreadPool

 **Answer:** B

---

### **20. Which improves responsiveness?**

- A) Blocking calls
- B) async/await
- C) Thread.Sleep
- D) lock everywhere

 **Answer:** B

---

## **MEDIUM LEVEL (20 MCQs – WITH SNIPPETS)**

### **21.**

```
Thread t = new Thread(() => Console.WriteLine("A"));
t.Start();
```

Output?

- A) A
- B) Error
- C) Nothing
- D) Depends

 **Answer:** A

---

### **22.**

```
Thread t = new Thread(M);  
t.Start(10);
```

M must match?

- A) ThreadStart
- B) Action
- C) Func
- D) ParameterizedThreadStart

 **Answer:** D

---

**23.**

```
void M(object x) {  
    Console.WriteLine(x);  
}
```

x is of type?

- A) int
- B) string
- C) object
- D) dynamic

 **Answer:** C

---

**24.**

```
ThreadPool.QueueUserWorkItem(_ => Console.WriteLine("T"));
```

Thread type?

- A) Foreground
- B) Background
- C) UI
- D) Main

 **Answer:** B

---

**25.**

```
lock(obj) {  
    x++;  
}
```

Purpose?

- A) Speed up code
- B) Synchronization
- C) Atomic operation
- D) Thread creation

 **Answer:** B

---

**26.**

```
Monitor.Enter(obj);  
try { }  
finally { Monitor.Exit(obj); }
```

Equivalent to?

- A) lock
- B) async
- C) await
- D) ThreadPool

 **Answer:** A

---

**27.**

```
Interlocked.Increment(ref x);
```

Ensures?

- A) Locking
- B) Atomic increment
- C) Thread creation
- D) Async execution

 **Answer:** B

---

**28.**

```
Task t = Task.Run(() => Console.WriteLine("X"));
```

Runs on?

- A) New thread
- B) ThreadPool thread

- C) UI thread
- D) Main thread

 **Answer:** B

---

**29.**

```
Task<int> t = Task.Run(() => 5);
Console.WriteLine(t.Result);
```

Output?

- A) 0
- B) 5
- C) Error
- D) Deadlock

 **Answer:** B

---

**30.**

```
async Task M() {
    await Task.Delay(1000);
}
```

Delay is:

- A) Blocking
- B) Non-blocking
- C) Sleep
- D) Lock

 **Answer:** B

---

**31.**

```
async void M() {}
```

Best used for?

- A) Business logic
- B) Library methods
- C) Event handlers
- D) Parallel loops

 **Answer:** C

---

**32.**

```
Task t = Task.Run(() => throw new Exception());
```

Exception observed when?

- A) Creation
- B) Execution
- C) Await/Result
- D) Never

 **Answer:** C

---

**33.**

```
await t;
```

If t throws exception?

- A) Ignored
- B) Wrapped
- C) Re-thrown
- D) Lost

 **Answer:** C

---

**34.**

```
Task.WaitAll(t1, t2);
```

Behavior?

- A) Async wait
- B) Parallel run
- C) Blocking wait
- D) Fire-and-forget

 **Answer:** C

---

**35.**

```
Parallel.For(0,10,i=>Console.WriteLine(i));
```

Execution?

- A) Sequential
- B) Parallel
- C) Async
- D) Single thread

 **Answer:** B

---

### **36. Which is TRUE?**

- A) Threads are lighter than Tasks
- B) Tasks use ThreadPool
- C) async creates new thread
- D) await blocks thread

 **Answer:** B

---

### **37.**

`Task t = Task.Run(M);`

M must return?

- A) void only
- B) Task only
- C) void or Task
- D) int

 **Answer:** C

---

### **38. Which avoids deadlocks?**

- A) Task.Result
- B) Thread.Sleep
- C) async/await
- D) lock

 **Answer:** C

---

### **39.**

`Task.Run(()=>M()).Wait();`

Problem?

- A) Faster
- B) Deadlock risk
- C) Async execution
- D) Safe

 **Answer:** B

---

#### **40. Which is thread-safe increment?**

- A) `x++`
- B) `lock + x++`
- C) `Interlocked.Increment`
- D) All

 **Answer:** D

---

### **HARD LEVEL (20 MCQs – TRICKY + SNIPPETS)**

**41.**

```
int x=0;  
Parallel.For(0,1000,i=>x++);
```

Value of x?

- A) 1000
- B) Less than 1000
- C) 0
- D) Exception

 **Answer:** B

---

**42.**

```
int x=0;  
Parallel.For(0,1000,i=>Interlocked.Increment(ref x));
```

Value?

- A) <1000
- B) 0
- C) 1000
- D) Random

**Answer:** C

---

**43.**

```
Task t = Task.Run(()=>Console.WriteLine("A"));
Console.WriteLine("B");
```

Output?

- A) AB
- B) BA
- C) Non-deterministic
- D) Error

**Answer:** C

---

**44.**

```
async Task<int> M() {
    return 5;
}
```

M returns?

- A) int
- B) Task
- C) Task<int>
- D) void

**Answer:** C

---

**45.**

```
var r = M();
Console.WriteLine(r.Result);
```

Effect?

- A) Async wait
- B) Blocking wait
- C) Compile error
- D) Deadlock safe

**Answer:** B

---

**46.**

```
lock(null) { }
```

Result?

- A) Valid
- B) Runtime exception
- C) Compile error
- D) Ignored

 **Answer:** B

---

**47. Which should be avoided?**

- A) Interlocked
- B) lock on private object
- C) lock on this
- D) Monitor

 **Answer:** C

---

**48.**

```
Task.Run(()=> { throw new Exception(); });
```

If not awaited?

- A) App crash
- B) Ignored silently
- C) Exception lost/logged later
- D) Compile error

 **Answer:** C

---

**49.**

```
async Task M() {
    await Task.Run(()=>{});
}
```

Threads used?

- A) One
- B) Two

- C) Many
- D) Zero

 **Answer:** B

---

## 50. Which is NOT true?

- A) async improves scalability
- B) await blocks thread
- C) Tasks are preferred
- D) ThreadPool threads are background

 **Answer:** B

---

## 51.

```
Parallel.Invoke(  
    ()=>Console.WriteLine("A"),  
    ()=>Console.WriteLine("B"));
```

Output?

- A) AB
- B) BA
- C) Non-deterministic
- D) Error

 **Answer:** C

---

## 52. Which is best for CPU-bound work?

- A) async/await
- B) Task.Run
- C) Thread.Sleep
- D) I/O async

 **Answer:** B

---

## 53. Which is best for I/O-bound work?

- A) Thread
- B) Task.Run

- A) async/await
- B) Parallel.For

 **Answer:** C

---

#### 54.

`Task t = Task.Run(()=>5);`

Compile?

- A) Yes
- B) No

 **Answer:** B

 (Return value must be consumed)

---

#### 55. Which ensures mutual exclusion?

- A) async
- B) await
- C) lock
- D) Task

 **Answer:** C

---

#### 56. Which is lightweight?

- A) Thread
- B) Process
- C) Task
- D) AppDomain

 **Answer:** C

---

#### 57. Which avoids race conditions?

- A) Parallel.For
- B) Interlocked
- C) ThreadPool
- D) async

 **Answer:** B

---

## **58. Which causes context switch?**

- A) Task.Delay
- B) Thread.Sleep
- C) await
- D) lock

 **Answer:** B

---

## **59. Which is MOST asked in CCEE?**

- A) Advanced TPL internals
- B) async/await behavior
- C) Thread priorities
- D) Unsafe code

 **Answer:** B

---

## **60. Which is BEST practice?**

- A) Prefer Thread
- B) Prefer Task + async/await
- C) Block with .Result
- D) Lock everywhere

 **Answer:** B

---

# **SESSION 13 – ASP.NET CORE MVC & ADO.NET (CCEE MCQs)**

---

## **EASY LEVEL (30 MCQs)**

### **1. ASP.NET Core MVC follows which pattern?**

- A) MVP
- B) MVVM
- C) MVC
- D) Layered

 Answer: C

---

## 2. MVC stands for:

- A) Model View Component
- B) Model View Controller
- C) Module View Controller
- D) Model Value Controller

 Answer: B

---

## 3. Which component handles user input?

- A) Model
- B) View
- C) Controller
- D) Middleware

 Answer: C

---

## 4. Which folder contains controllers?

- A) Views
- B) Models
- C) Controllers
- D) Pages

 Answer: C

---

## 5. Which file contains app configuration in ASP.NET Core?

- A) web.config
- B) appsettings.json
- C) global.asax
- D) startup.xml

 Answer: B

---

## **6. Which method configures middleware?**

- A) ConfigureServices
- B) Configure
- C) Main
- D) Init

 **Answer:** B

---

## **7. Controllers must inherit from:**

- A) Object
- B) Controller
- C) BaseController
- D) IAction

 **Answer:** B

---

## **8. Which suffix is required for MVC controllers?**

- A) Ctrl
- B) Controller
- C) Api
- D) Action

 **Answer:** B

---

## **9. An action method must be:**

- A) private
- B) static
- C) public
- D) protected

 **Answer:** C

---

## **10. Which attribute maps HTTP GET?**

- A) [Get]
- B) [HttpGet]

- C) [HttpPost]
- D) [GetAction]

 **Answer:** B

---

### **11. Which attribute maps HTTP POST?**

- A) [Post]
- B) [HttpPost]
- C) [HttpPut]
- D) [PostAction]

 **Answer:** B

---

### **12. Which attribute prevents a method from being an action?**

- A) [NoAction]
- B) [Ignore]
- C) [NonAction]
- D) [Disable]

 **Answer:** C

---

### **13. Which method returns a View?**

- A) return View()
- B) return Page()
- C) return Result()
- D) return Html()

 **Answer:** A

---

### **14. Default route maps to which action?**

- A) Create
- B) Index
- C) Home
- D) Main

 **Answer:** B

---

## **15. Which namespace contains SqlConnection?**

- A) System.Data
- B) System.Data.SqlClient
- C) Microsoft.Data.SqlClient
- D) System.Sql

 **Answer:** C

---

## **16. SqlConnection is used to:**

- A) Execute query
- B) Hold data
- C) Connect to database
- D) Read data

 **Answer:** C

---

## **17. Which object executes SQL commands?**

- A) SqlDataReader
- B) SqlCommand
- C) SqlDataAdapter
- D) DataSet

 **Answer:** B

---

## **18. Which object reads data forward-only?**

- A) DataSet
- B) DataTable
- C) SqlDataReader
- D) DataAdapter

 **Answer:** C

---

## **19. Which object works in disconnected mode?**

- A) DataReader
- B) SqlCommand
- C) DataSet
- D) SqlConnection

 Answer: C

---

## 20. Which supports CRUD operations?

- A) SqlConnection
- B) SqlCommand
- C) DataSet
- D) All

 Answer: D

---

## 21. Which method opens DB connection?

- A) Connect()
- B) Start()
- C) Open()
- D) Begin()

 Answer: C

---

## 22. Which closes connection?

- A) Close()
- B) Dispose()
- C) Both A & B
- D) End()

 Answer: C

---

## 23. Which file contains routing info?

- A) Startup.cs
- B) Program.cs
- C) appsettings.json
- D) web.config

 Answer: A

---

## 24. Which is asynchronous DB method?

- A) ExecuteReader()
- B) ExecuteNonQuery()
- C) ExecuteReaderAsync()
- D) Fill()

 **Answer:** C

---

## **25. MVC View is written using:**

- A) HTML only
- B) Razor
- C) JavaScript
- D) CSS

 **Answer:** B

---

## **26. Which folder contains Razor views?**

- A) Models
- B) Views
- C) Controllers
- D) Pages

 **Answer:** B

---

## **27. Which is TRUE?**

- A) Controller directly accesses DB
- B) View handles logic
- C) Model represents data
- D) MVC has no routing

 **Answer:** C

---

## **28. Which action result returns JSON?**

- A) ViewResult
- B) ContentResult
- C) JsonResult
- D) FileResult

 **Answer:** C

---

## 29. Which attribute defines route?

- A) [Path]
- B) [Route]
- C) [Url]
- D) [Map]

 Answer: B

---

## 30. Which is preferred in ASP.NET Core?

- A) Synchronous DB calls
- B) Asynchronous DB calls
- C) Blocking threads
- D) Static connections

 Answer: B

---

# MEDIUM LEVEL (30 MCQs – MANY SNIPPETS)

### 31.

```
public class HomeController : Controller {}
```

This class is:

- A) Model
- B) View
- C) Controller
- D) Middleware

 Answer: C

---

### 32.

```
public IActionResult Index() {
    return View();
}
```

Returns?

- A) HTML
- B) ViewResult
- C) string
- D) object

 **Answer:** B

---

**33.**

```
[HttpGet]  
public IActionResult Create(){}  
  
Invoked by?
```

- A) POST
- B) GET
- C) PUT
- D) DELETE

 **Answer:** B

---

**34.**

```
[HttpPost]  
public IActionResult Create(){}  
  
Invoked by?
```

- A) GET
- B) POST
- C) PUT
- D) PATCH

 **Answer:** B

---

**35.**

```
[NonAction]  
public void Helper(){}  
  
Helper is:
```

- A) Action
- B) Private

- C) Not callable as action
- D) Middleware

 **Answer:** C

---

**36.**

```
return RedirectToAction("Index");
```

This does?

- A) Returns view
- B) Redirects request
- C) Calls method directly
- D) Renders partial

 **Answer:** B

---

**37.**

```
return Content("Hello");
```

Returns?

- A) View
- B) JSON
- C) Plain text
- D) File

 **Answer:** C

---

**38.**

```
SqlConnection con = new SqlConnection(cs);
```

cs is?

- A) SQL
- B) Query
- C) Connection string
- D) Adapter

 **Answer:** C

---

**39.**

```
SqlCommand cmd = new SqlCommand(sql, con);
```

Purpose?

- A) Open DB
- B) Execute query
- C) Read data
- D) Close DB

 **Answer:** B

---

**40.**

```
SqlDataReader dr = cmd.ExecuteReader();
```

Mode?

- A) Disconnected
- B) Connected
- C) Cached
- D) Offline

 **Answer:** B

---

**41.**

```
DataSet ds = new DataSet();
```

Nature?

- A) Connected
- B) Forward only
- C) Disconnected
- D) Async

 **Answer:** C

---

**42.**

```
SqlDataAdapter da = new SqlDataAdapter(cmd);
```

Used for?

- A) Execute query
- B) Fill DataSet

- C) Read stream
- D) Async call

 **Answer:** B

---

**43.**

```
await cmd.ExecuteNonQuery();
```

This is:

- A) Blocking
- B) Async
- C) Sync
- D) Parallel

 **Answer:** B

---

**44.**

```
using(SqlConnection con = new SqlConnection(cs)){}  
    
```

Ensures?

- A) Faster execution
- B) Automatic Dispose
- C) Thread safety
- D) Caching

 **Answer:** B

---

**45.**

```
return Json(obj);  
    
```

Returns?

- A) View
- B) HTML
- C) JSON
- D) XML

 **Answer:** C

---

**46.**

```
[Route("home/index")]
```

Defines?

- A) Action
- B) Model
- C) URL mapping
- D) View name

 **Answer:** C

---

**47.**

```
public IActionResult Details(int id){}
```

id comes from?

- A) View
- B) Route / Query
- C) Database
- D) Model

 **Answer:** B

---

**48.**

```
cmd.Parameters.AddWithValue("@id", 1);
```

Used for?

- A) Security
- B) Performance
- C) Parameterized query
- D) All

 **Answer:** D

---

**49.**

```
SqlCommand cmd;  
cmd.ExecuteScalar();
```

Returns?

- A) Row
- B) Table
- C) Single value
- D) Reader

 **Answer:** C

---

**50.**

`ExecuteReaderAsync()`

Returns?

- A) Task
- B) Task<int>
- C) Task<SqlDataReader>
- D) SqlDataReader

 **Answer:** C

---

**51.**

`services.AddControllersWithViews();`

Registers?

- A) Models
- B) MVC services
- C) Routing
- D) DB

 **Answer:** B

---

**52.**

`app.UseRouting();`

Used for?

- A) DB routing
- B) MVC routing
- C) File routing
- D) Network routing

 **Answer:** B

---

**53.**

```
return View(model);
```

Purpose?

- A) Pass data to view
- B) Save model
- C) Validate model
- D) Bind model

 **Answer:** A

---

**54. Which prevents SQL Injection?**

- A) Dynamic SQL
- B) String concat
- C) Parameters
- D) Views

 **Answer:** C

---

**55. Which is faster for large reads?**

- A) DataSet
- B) DataReader
- C) DataTable
- D) LINQ

 **Answer:** B

---

**56. Which supports async DB connection?**

- A) Open()
- B) OpenAsync()
- C) Connect()
- D) Start()

 **Answer:** B

---

**57. Which returns HTTP status?**

- A) StatusCodeResult
- B) ViewResult
- C) JsonResult
- D) ContentResult

 **Answer:** A

---

### 58.

```
return NotFound();
```

Returns?

- A) 200
- B) 400
- C) 404
- D) 500

 **Answer:** C

---

### 59. Which is TRUE?

- A) MVC is stateful
- B) Controllers are singleton
- C) MVC is stateless
- D) Views handle logic

 **Answer:** C

---

### 60. Which is best practice?

- A) DB logic in Controller
- B) Async DB calls
- C) Hardcoded SQL
- D) Blocking threads

 **Answer:** B

---

 **HARD LEVEL (30 MCQs – TRICKY + SNIPPETS)**

**61.**

```
public IActionResult Index(){
    return RedirectToAction("Index");
}
```

Result?

- A) Infinite loop
- B) Compile error
- C) Runtime error
- D) View rendered

 **Answer:** A

---

**62.**

```
SqlDataReader dr;
con.Close();
dr.Read();
```

Result?

- A) Works
- B) Runtime exception
- C) Compile error
- D) No data

 **Answer:** B

---

**63.**

```
await con.OpenAsync();
await cmd.ExecuteReaderAsync();
```

Connection state?

- A) Closed
- B) Open
- C) Disposed
- D) Unknown

 **Answer:** B

---

**64.**

```
using(SqlConnection con){}
```

If exception?

- A) Leak
- B) Dispose still called
- C) Connection open
- D) App crash

 **Answer:** B

---

**65.**

```
return View("Create");
```

Renders?

- A) Index
- B) Create view
- C) Action
- D) Partial

 **Answer:** B

---

**66.**

```
public IActionResult Edit(){  
    return View();  
    return Json(obj);  
}
```

Result?

- A) Both executed
- B) First return only
- C) Compile error
- D) Runtime error

 **Answer:** B

---

**67.**

```
cmd.ExecuteNonQuery();
```

Used for?

- A) SELECT
- B) INSERT/UPDATE/DELETE
- C) READ
- D) JOIN

 **Answer:** B

---

**68.**

```
DataSet ds;  
ds.Tables[0];
```

Requires?

- A) DataReader
- B) DataAdapter.Fill()
- C) Connection open
- D) Async

 **Answer:** B

---

**69.**

```
ExecuteScalarAsync()
```

Returns?

- A) Task
- B) Task<object>
- C) object
- D) int

 **Answer:** B

---

**70. Which causes deadlock risk?**

- A) Async controller
- B) await OpenAsync
- C) .Result on async
- D) Parameterized SQL

 **Answer:** C

---

**71.**

[HttpGet,HttpPost]

Effect?

- A) Error
- B) Both allowed
- C) Only GET
- D) Only POST

 **Answer:** B

---

**72.**

```
public IActionResult Index(int? id){}
```

If id missing?

- A) Exception
- B) null
- C) 0
- D) Compile error

 **Answer:** B

---

**73.**

```
return PartialView();
```

Used for?

- A) Full page
- B) Partial rendering
- C) JSON
- D) Redirect

 **Answer:** B

---

**74. Which is NOT MVC responsibility?**

- A) Routing
- B) Business logic
- C) Data rendering
- D) Threading

 **Answer:** D

---

**75.**

```
cmd.CommandType = CommandType.StoredProcedure;
```

Executes?

- A) SQL
- B) Table
- C) Stored procedure
- D) View

 **Answer:** C

---

**76. Which object is NOT thread-safe?**

- A) SqlConnection
- B) SqlCommand
- C) DataReader
- D) All

 **Answer:** D

---

**77.**

```
await cmd.ExecuteReaderAsync();
```

If not awaited?

- A) Error
- B) Fire and forget
- C) Exception lost
- D) All

 **Answer:** D

---

**78. Which improves scalability?**

- A) Synchronous MVC
- B) Async MVC
- C) Static DB calls
- D) Locks

 **Answer:** B

---

## 79. Which returns HTTP 500?

- A) Ok()
- B) BadRequest()
- C) StatusCode(500)
- D) NotFound()

 **Answer:** C

---

## 80.

```
return View();
return RedirectToAction("Index");
```

Execution?

- A) Both
- B) First only
- C) Second only
- D) Error

 **Answer:** B

---

## 81. Which maps URL to controller?

- A) Model
- B) Routing
- C) View engine
- D) Middleware

 **Answer:** B

---

## 82. Which is NOT async?

- A) OpenAsync
- B) ExecuteReaderAsync
- C) Fill
- D) ExecuteNonQueryAsync

 **Answer:** C

---

### **83. Which is best for high load?**

- A) DataSet
- B) DataReader + async
- C) Sync calls
- D) Blocking threads

 **Answer:** B

---

### **84. Which is stateless?**

- A) MVC controller
- B) View
- C) Model
- D) DB connection

 **Answer:** A

---

### **85. Which improves security?**

- A) Inline SQL
- B) Stored procedures
- C) Parameters
- D) B + C

 **Answer:** D

---

### **86. Which breaks MVC?**

- A) Thin controller
- B) Fat controller
- C) View models
- D) Separation

 **Answer:** B

---

### **87. Which action result is lightweight?**

- A) ViewResult
- B) JsonResult

- C) ContentResult
- D) FileResult

 **Answer:** C

---

### **88. Which is most asked in CCEE?**

- A) Razor syntax
- B) MVC architecture flow
- C) Middleware internals
- D) HTML helpers

 **Answer:** B

---

### **89. Which is recommended?**

- A) DB in View
- B) Async Controller + Async DB
- C) Sync everywhere
- D) Static SqlConnection

 **Answer:** B

---

### **90. Which gives MAX marks in CCEE?**

- A) Writing full MVC app
- B) Understanding flow + snippets
- C) UI design
- D) CSS

 **Answer:** B

## **SESSIONS 14 & 15 – MVC VIEWS, MODELS, VALIDATION & STATE (CCEE MCQs)**

---

 **EASY LEVEL (25 MCQs)**

## **1. A Model in MVC represents:**

- A) UI
- B) Business logic & data
- C) Routing
- D) Middleware

 **Answer:** B

---

## **2. A ViewModel is used to:**

- A) Store DB connection
- B) Combine multiple models for view
- C) Replace controller
- D) Handle routing

 **Answer:** B

---

## **3. Razor views use which extension?**

- A) .html
- B) .cs
- C) .cshtml
- D) .razor

 **Answer:** C

---

## **4. Which symbol starts Razor code?**

- A) #
- B) @
- C) \$
- D) %

 **Answer:** B

---

## **5. HTML Helper functions are used to:**

- A) Access DB
- B) Generate HTML
- C) Handle routing
- D) Manage sessions

 **Answer:** B

---

## **6. Which helper generates textbox?**

- A) Html.Text()
- B) Html.TextBox()
- C) Html.TextBoxFor()
- D) Html.Input()

 **Answer:** C

---

## **7. ViewBag is:**

- A) Strongly typed
- B) Dynamic
- C) Static
- D) Read-only

 **Answer:** B

---

## **8. ViewBag data lifetime is:**

- A) Application lifetime
- B) Session lifetime
- C) Single request
- D) Browser lifetime

 **Answer:** C

---

## **9. Which attribute makes field mandatory?**

- A) [Key]
- B) [Required]
- C) [Range]
- D) [StringLength]

 **Answer:** B

---

## **10. Data Annotations are used for:**

- A) Routing
- B) Validation
- C) Authorization
- D) Logging

 **Answer:** B

---

### **11. Client-side validation runs on:**

- A) Server
- B) Browser
- C) Database
- D) Controller

 **Answer:** B

---

### **12. Server-side validation runs on:**

- A) Browser
- B) Client
- C) Controller
- D) Database

 **Answer:** C

---

### **13. Strongly typed views are declared using:**

- A) @model
- B) @using
- C) @inject
- D) @view

 **Answer:** A

---

### **14. Scaffold templates help to:**

- A) Write SQL
- B) Auto-generate CRUD UI
- C) Create controllers only
- D) Handle security

 **Answer:** B

---

## **15. Which object passes data between requests?**

- A) ViewBag
- B) ViewData
- C) TempData
- D) Model

 **Answer:** C

---

## **16. TempData uses:**

- A) Cookies
- B) Session internally
- C) QueryString
- D) Application state

 **Answer:** B

---

## **17. Session data is stored:**

- A) Client side
- B) Server side
- C) View
- D) Query string

 **Answer:** B

---

## **18. Which is client-side state management?**

- A) Session
- B) TempData
- C) Cookies
- D) Application

 **Answer:** C

---

## **19. QueryString data is:**

- A) Hidden
- B) Encrypted

- C) Visible in URL
- D) Server only

 **Answer:** C

---

## **20. Partial View is used to:**

- A) Render full page
- B) Reuse UI components
- C) Handle routing
- D) Access DB

 **Answer:** B

---

## **21. Partial views start with:**

- A) \$
- B) \_
- C) @
- D) #

 **Answer:** B

---

## **22. Which returns partial HTML?**

- A) View()
- B) PartialView()
- C) Json()
- D) Content()

 **Answer:** B

---

## **23. Application state is:**

- A) Per request
- B) Per user
- C) Shared across users
- D) Browser based

 **Answer:** C

---

## 24. Which is NOT state management?

- A) ViewBag
- B) TempData
- C) Session
- D) LINQ

 Answer: D

---

## 25. Which is fastest?

- A) Session
- B) ViewBag
- C) TempData
- D) Database

 Answer: B

---

## MEDIUM LEVEL (25 MCQs – MANY SNIPPETS)

### 26.

`@ViewBag .Name`

ViewBag is:

- A) Strongly typed
- B) Dynamic
- C) Static
- D) Read-only

 Answer: B

---

### 27.

`@model Employee`

Meaning?

- A) Model declaration
- B) DB mapping
- C) Routing
- D) Validation

 Answer: A

---

**28.**

```
@Html.TextBoxFor(m => m.Name)
```

Generates?

- A) Label
- B) TextBox
- C) Dropdown
- D) Hidden field

 **Answer:** B

---

**29.**

```
[Required]
```

```
public string Name { get; set; }
```

Validation?

- A) Range
- B) Mandatory
- C) Length
- D) Regex

 **Answer:** B

---

**30.**

```
if(ModelState.IsValid)
```

Checks?

- A) Routing
- B) Authorization
- C) Validation
- D) Session

 **Answer:** C

---

**31.**

```
@Html.ValidationMessageFor(m => m.Name)
```

Displays?

- A) Label
- B) Error message
- C) Tooltip
- D) Placeholder

 **Answer:** B

---

**32.**

```
 TempData[ "msg" ] = "Saved";
```

Accessible?

- A) Same request only
- B) Next request only
- C) Always
- D) Never

 **Answer:** B

---

**33.**

```
 Session[ "user" ] = "admin";
```

Stored?

- A) Client side
- B) Server side
- C) View
- D) URL

 **Answer:** B

---

**34.**

```
@Html.Partial( "_Menu" )
```

Renders?

- A) View
- B) Partial view
- C) Layout
- D) Controller

 **Answer:** B

---

**35.**

```
return PartialView("_List", model);
```

Used for?

- A) Full page
- B) AJAX response
- C) Redirect
- D) Error page

 **Answer:** B

---

**36.**

```
@Html.BeginForm()
```

Creates?

- A) Anchor tag
- B) Form tag
- C) Input tag
- D) Button

 **Answer:** B

---

**37.**

```
[Range(1,100)]
```

Validation type?

- A) Length
- B) Numeric range
- C) Required
- D) Pattern

 **Answer:** B

---

**38.**

```
[StringLength(10)]
```

Restricts?

- A) Numeric value
- B) Text length
- C) Regex
- D) Null

 **Answer:** B

---

**39.**

```
public IActionResult Edit(int id)
```

id comes from?

- A) Model
- B) ViewBag
- C) Route/QueryString
- D) Session

 **Answer:** C

---

**40.**

```
@Html.DisplayFor(m => m.Name)
```

Used for?

- A) Edit
- B) Display only
- C) Validation
- D) Hidden

 **Answer:** B

---

**41.**

```
 ViewData["x"] = 10;
```

ViewData is:

- A) Dynamic
- B) Strongly typed
- C) Dictionary
- D) Static

 **Answer:** C

---

**42.**

```
return RedirectToAction("Index");
```

Effect?

- A) Renders view
- B) New request
- C) Partial render
- D) JSON

 **Answer:** B

---

**43.**

```
@Html.HiddenFor(m => m.Id)
```

Used for?

- A) UI display
- B) Storing value without display
- C) Validation
- D) Session

 **Answer:** B

---

**44.**

```
ModelState.AddModelError("", "Error");
```

Purpose?

- A) Logging
- B) Validation error
- C) Exception
- D) Redirect

 **Answer:** B

---

**45.**

```
@Html.ValidationSummary()
```

Displays?

- A) Field errors
- B) All errors
- C) One error
- D) Warning

 **Answer:** B

---

#### **46. Which persists longest?**

- A) ViewBag
- B) TempData
- C) Session
- D) QueryString

 **Answer:** C

---

#### **47. Which is stateless?**

- A) Session
- B) ViewBag
- C) Cookies
- D) Application

 **Answer:** B

---

#### **48. Which is safest?**

- A) QueryString
- B) Cookies
- C) Session
- D) ViewBag

 **Answer:** C

---

#### **49. Which enables CRUD quickly?**

- A) Razor
- B) Scaffold
- C) ViewBag
- D) Session

 **Answer:** B

---

## 50. Which validation runs always?

- A) Client-side
- B) Server-side
- C) JavaScript
- D) Browser

 Answer: B

---

## HARD LEVEL (25 MCQs – TRICKY + SNIPPETS)

### 51.

```
ViewBag.X = 10;  
return RedirectToAction("Index");
```

X available in Index?

- A) Yes
- B) No

 Answer: B

---

### 52.

```
TempData["msg"] = "Done";  
return RedirectToAction("Index");
```

msg available?

- A) Yes
- B) No

 Answer: A

---

### 53.

```
Session.Clear();
```

Effect?

- A) Clears one key
- B) Clears all session data

- C) Clears cookies
- D) Clears ViewBag

 **Answer:** B

---

**54.**

```
@Html.Partial("_View")  
@Html.Partial("_View")
```

Rendered?

- A) Once
- B) Twice
- C) Error
- D) Cached

 **Answer:** B

---

**55.**

```
@model IEnumerable<Employee>
```

Used for?

- A) Single object
- B) Collection
- C) ViewBag
- D) Partial

 **Answer:** B

---

**56.**

```
@if(ViewBag.X > 5) { }
```

ViewBag.X missing?

- A) Exception
- B) Compile error
- C) Runtime error
- D) Null check fails

 **Answer:** C

---

**57.**

```
public class VM {  
    public Employee Emp {get;set;}  
    public List<Role> Roles {get;set;}  
}
```

VM is:

- A) Model
- B) ViewModel
- C) Entity
- D) DTO

 **Answer:** B

---

**58.**

```
[ValidateAntiForgeryToken]
```

Used for?

- A) Validation
- B) Security
- C) Routing
- D) Session

 **Answer:** B

---

**59.**

```
@Html.AntiForgeryToken()
```

Protects against?

- A) SQL Injection
- B) XSS
- C) CSRF
- D) DOS

 **Answer:** C

---

**60.**

```
public IActionResult Create(Employee e)
```

e is populated using?

- A) ViewBag
- B) Model binding
- C) Session
- D) TempData

 **Answer:** B

---

**61.**

`@Html.EditorFor(m => m.Name)`

Generates?

- A) Label
- B) Input based on type
- C) Static text
- D) Hidden field

 **Answer:** B

---

**62.**

`Application["count"]++;`

Risk?

- A) Safe
- B) Thread safety issue
- C) Compile error
- D) Null

 **Answer:** B

---

**63.**

`QueryString: ?id=5`

Risk?

- A) Secure
- B) Tampering
- C) Encrypted
- D) Hidden

 **Answer:** B

---

**64.**

```
return View();  
return PartialView();
```

Execution?

- A) Both
- B) First only
- C) Second only
- D) Error

 **Answer:** B

---

**65.**

```
@Html.RenderPartial("_View")
```

Difference?

- A) Returns string
- B) Writes directly to response
- C) Same as Partial
- D) Async

 **Answer:** B

---

**66. Which is NOT persistent?**

- A) Cookies
- B) Session
- C) ViewBag
- D) TempData

 **Answer:** C

---

**67. Which breaks MVC?**

- A) Strongly typed views
- B) Fat views
- C) ViewModels
- D) Partial views

 **Answer:** B

---

## **68. Which validation cannot be bypassed?**

- A) Client-side
- B) JavaScript
- C) Server-side
- D) Browser

 **Answer:** C

---

## **69. Which is fastest state passing?**

- A) Session
- B) TempData
- C) ViewBag
- D) Cookies

 **Answer:** C

---

## **70. Which is best practice?**

- A) Heavy ViewBag usage
- B) Strongly typed views
- C) Logic in view
- D) DB access in view

 **Answer:** B

---

## **71.**

```
@Html.ActionLink("Edit", "Edit", new{id=1})
```

Generates?

- A) Form
- B) Button
- C) Anchor tag
- D) Script

 **Answer:** C

---

## **72. Which is shared across users?**

- A) Session
- B) ViewBag
- C) Application
- D) TempData

 **Answer:** C

---

## **73. Which is recommended for CRUD?**

- A) ViewBag
- B) ViewModel
- C) Session
- D) QueryString

 **Answer:** B

---

## **74. Which is MOST asked in CCEE?**

- A) Razor syntax details
- B) ViewBag vs TempData vs Session
- C) CSS in MVC
- D) Bootstrap

 **Answer:** B

---

## **75. Which gives MAX score?**

- A) Writing views
- B) Predicting MVC flow & snippets
- C) UI design
- D) Styling

 **Answer:** B

 **SESSIONS 16 & 17 – ROUTING, MVC  
LIFECYCLE, LAYOUTS, SECURITY (CCEE MCQs)**

---

## EASY LEVEL (30 MCQs)

### 1. Routing in MVC is used to:

- A) Secure application
- B) Map URL to controller/action
- C) Validate input
- D) Handle database

 Answer: B

---

### 2. Routing engine uses:

- A) Controller
- B) Routing table
- C) View engine
- D) Middleware only

 Answer: B

---

### 3. Default MVC route usually maps to:

- A) Home/Create
- B) Home/Index
- C) Default/Main
- D) App/Start

 Answer: B

---

### 4. Which file configures conventional routing?

- A) Program.cs
- B) RouteConfig.cs
- C) Startup.cs
- D) web.config

 Answer: C

---

### 5. A 404 error indicates:

- A) Server error
- B) Unauthorized
- C) Resource not found
- D) Bad request

 **Answer:** C

---

## **6. Attribute routing is enabled using:**

- A) UseMvc()
- B) UseRouting()
- C) MapControllers()
- D) MapAttributeRoutes()

 **Answer:** D

---

## **7. Which attribute defines route on action?**

- A) [HttpGet]
- B) [Route]
- C) [Action]
- D) [Url]

 **Answer:** B

---

## **8. Request Life Cycle starts with:**

- A) Controller
- B) Routing
- C) View
- D) Model

 **Answer:** B

---

## **9. Middleware executes in:**

- A) Random order
- B) Reverse order
- C) Pipeline order
- D) Parallel

 **Answer:** C

---

## **10. Layout in MVC is similar to:**

- A) Partial View
- B) Master Page
- C) Controller
- D) Model

 **Answer:** B

---

## **11. Layout file extension is:**

- A) .layout
- B) .cs
- C) .cshtml
- D) .html

 **Answer:** C

---

## **12. Default layout file is:**

- A) \_Layout.cs
- B) Layout.cshtml
- C) \_Layout.cshtml
- D) MainLayout.cshtml

 **Answer:** C

---

## **13. Bundling is used to:**

- A) Minimize DB calls
- B) Combine CSS/JS
- C) Secure application
- D) Route URLs

 **Answer:** B

---

## **14. Minification does:**

- A) Compress images
- B) Remove whitespace/comments

- C) Encrypt code
- D) Bundle files

 **Answer:** B

---

### **15. BundleConfig file is used for:**

- A) Routing
- B) Security
- C) Bundling & minification
- D) Validation

 **Answer:** C

---

### **16. Which file contains CSS links?**

- A) Controller
- B) Layout
- C) Model
- D) RouteConfig

 **Answer:** B

---

### **17. Custom helper functions are used to:**

- A) Access DB
- B) Generate reusable HTML
- C) Handle routing
- D) Secure data

 **Answer:** B

---

### **18. Async actions improve:**

- A) UI design
- B) Scalability
- C) Routing
- D) Security

 **Answer:** B

---

## **19. Which keyword makes action async?**

- A) await
- B) async
- C) Task
- D) parallel

 **Answer:** B

---

## **20. Which returns async result?**

- A) IActionResult
- B) Task<ActionResult>
- C) ActionResult
- D) JsonResult

 **Answer:** B

---

## **21. Filters are used to:**

- A) Format data
- B) Execute logic before/after actions
- C) Manage state
- D) Route URLs

 **Answer:** B

---

## **22. Authorization filter runs:**

- A) After action
- B) Before action
- C) After result
- D) After exception

 **Answer:** B

---

## **23. Which attribute restricts access?**

- A) [AllowAnonymous]
- B) [Authorize]
- C) [ValidateInput]
- D) [HttpPost]

 **Answer:** B

---

#### **24. Which allows public access?**

- A) [Authorize]
- B) [AllowAnonymous]
- C) [Public]
- D) [Open]

 **Answer:** B

---

#### **25. Forms authentication uses:**

- A) Session
- B) Cookies
- C) ViewBag
- D) QueryString

 **Answer:** B

---

#### **26. AntiForgeryToken prevents:**

- A) SQL Injection
- B) CSRF
- C) XSS
- D) DOS

 **Answer:** B

---

#### **27. Cross Site Scripting attack targets:**

- A) Database
- B) Browser
- C) Server memory
- D) Network

 **Answer:** B

---

#### **28. HTML encoding prevents:**

- A) CSRF
- B) SQL Injection
- C) XSS
- D) Routing issues

 Answer: C

---

## 29. MVC security is implemented using:

- A) Filters & attributes
- B) Views
- C) Models
- D) LINQ

 Answer: A

---

## 30. Which is MOST important for CCEE?

- A) UI design
- B) Routing & security flow
- C) CSS styling
- D) JavaScript

 Answer: B

---

## MEDIUM LEVEL (30 MCQs – MANY SNIPPETS)

### 31.

```
routes.MapRoute(  
    name:"default",  
    pattern:"{controller=Home}/{action=Index}/{id?}"  
)
```

Purpose?

- A) Attribute routing
- B) Conventional routing
- C) API routing
- D) Error handling

 Answer: B

---

**32.**

```
[Route("products/list")]
```

Maps to:

- A) Controller
- B) Action URL
- C) Layout
- D) View

 **Answer:** B

---

**33.**

```
[HttpGet]  
public IActionResult Index(){}
```

Triggered by?

- A) POST
- B) GET
- C) PUT
- D) DELETE

 **Answer:** B

---

**34.**

```
[Route("home")]  
[Route("home/index")]
```

Effect?

- A) Error
- B) Multiple URLs
- C) Override
- D) Ignore

 **Answer:** B

---

**35.**

```
return NotFound();
```

Returns HTTP?

- A) 200
- B) 401
- C) 404
- D) 500

 **Answer:** C

---

**36.**

`@RenderBody()`

Used for?

- A) Partial view
- B) View content placeholder
- C) Scripts
- D) Styles

 **Answer:** B

---

**37.**

`@RenderSection("Scripts", required:false)`

Used for?

- A) CSS
- B) JS section
- C) Layout logic
- D) Routing

 **Answer:** B

---

**38.**

`bundles.Add(new ScriptBundle("~/bundles/jquery"));`

Adds?

- A) CSS
- B) JS bundle
- C) Layout
- D) Route

 **Answer:** B

---

**39.**

```
@Styles.Render("~/Content/css")
```

Renders?

- A) JS
- B) CSS
- C) HTML
- D) Partial

 **Answer:** B

---

**40.**

```
@Scripts.Render("~/bundles/bootstrap")
```

Used for?

- A) Validation
- B) Script loading
- C) Routing
- D) Model binding

 **Answer:** B

---

**41.**

```
public static MvcHtmlString MyHelper()
```

This is:

- A) Controller
- B) Custom helper
- C) Filter
- D) Route

 **Answer:** B

---

**42.**

```
public async Task<IActionResult> Index()
```

This is:

- A) Sync action
- B) Async action
- C) Parallel action
- D) Background task

 **Answer:** B

---

**43.**

```
await Task.Delay(1000);
```

Effect?

- A) Blocks thread
- B) Frees thread
- C) Sleeps thread
- D) Locks thread

 **Answer:** B

---

**44.**

```
try { } catch { Log(); }
```

Purpose?

- A) Routing
- B) Error logging
- C) Validation
- D) Authorization

 **Answer:** B

---

**45.**

```
public class LogFilter : IActionFilter
```

This is:

- A) Exception filter
- B) Action filter
- C) Result filter
- D) Auth filter

 **Answer:** B

---

**46.**

OnActionExecuting()

Called:

- A) After action
- B) Before action
- C) After result
- D) On exception

 **Answer:** B

---

**47.**

```
[Authorize]  
public IActionResult Dashboard(){}  
 
```

Access?

- A) Everyone
- B) Authenticated users
- C) Admin only
- D) Anonymous

 **Answer:** B

---

**48.**

```
[AllowAnonymous]  
 
```

Overrides?

- A) Routing
- B) Validation
- C) Authorization
- D) Filters

 **Answer:** C

---

**49.**

```
@Html.AntiForgeryToken()  
 
```

Used with?

- A) GET
- B) POST
- C) PUT
- D) DELETE

 **Answer:** B

---

**50.**

`[ValidateAntiForgeryToken]`

Validates?

- A) Cookies
- B) CSRF token
- C) Session
- D) ViewBag

 **Answer:** B

---

**51.**

`@Html.Raw(" <script> ")`

Risk?

- A) SQL injection
- B) XSS
- C) CSRF
- D) Deadlock

 **Answer:** B

---

**52.**

`@Html.Encode(userInput)`

Prevents?

- A) SQL injection
- B) XSS
- C) CSRF
- D) Routing

 **Answer:** B

---

### **53. Which filter handles errors?**

- A) Authorization
- B) Action
- C) Exception
- D) Result

 **Answer:** C

---

### **54.**

```
app.UseEndpoints(endpoints => endpoints.MapControllers());
```

Purpose?

- A) Bundling
- B) Routing
- C) Security
- D) Logging

 **Answer:** B

---

### **55. Which executes LAST?**

- A) Authorization filter
- B) Action filter
- C) Result filter
- D) Exception filter

 **Answer:** C

---

### **56. Which improves performance?**

- A) Bundling
- B) Minification
- C) Both
- D) None

 **Answer:** C

---

### **57. Which is SEO friendly?**

- A) QueryString routing
- B) Attribute routing
- C) Random routing
- D) Static routing

 Answer: B

---

### 58. Which is async-safe?

- A) Thread.Sleep
- B) Task.Delay
- C) lock
- D) Wait()

 Answer: B

---

### 59. Which is MVC module?

- A) Partial view
- B) Action method
- C) Controller
- D) All

 Answer: D

---

### 60. Which is MOST tested in CCEE?

- A) Routing flow
- B) CSS bundles
- C) Bootstrap
- D) JavaScript

 Answer: A

---

## HARD LEVEL (30 MCQs – TRICKY + SNIPPETS)

61.

```
[Authorize]  
[AllowAnonymous]
```

Effective access?

- A) Authorized
- B) Anonymous
- C) Error
- D) Depends

 **Answer:** B

---

**62.**

```
routes.MapRoute("{*url}", new {controller="Home", action="Error"});
```

Used for?

- A) Default route
- B) Catch-all
- C) Attribute route
- D) API route

 **Answer:** B

---

**63.**

```
return RedirectToAction("Index");
```

Request type?

- A) Same request
- B) New request
- C) AJAX
- D) Partial

 **Answer:** B

---

**64.**

```
@RenderSection("Scripts", required:true)
```

If missing?

- A) Ignored
- B) Compile error
- C) Runtime error
- D) Warning

 **Answer:** C

---

**65.**

```
bundles.IgnoreList.Clear();
```

Effect?

- A) Ignore all files
- B) Use all files
- C) Error
- D) Disable bundling

 **Answer:** B

---

**66.**

```
public async Task<IActionResult> Index(){  
    return View();  
}
```

Async benefit?

- A) None
- B) Scalability
- C) Faster CPU
- D) Parallel UI

 **Answer:** B

---

**67.**

```
Task<IActionResult> Index()
```

Without `async` keyword?

- A) Valid
- B) Compile error
- C) Runtime error
- D) Deadlock

 **Answer:** A

---

**68.**

```
HandleErrorAttribute
```

Used for?

- A) Logging only
- B) Exception handling
- C) Routing
- D) Validation

 **Answer:** B

---

**69.**

`@Html.DisplayFor(m=>m.Name)`

Encodes output?

- A) Yes
- B) No

 **Answer:** A

---

**70.**

`@Html.Raw(Model.Name)`

Risk?

- A) SQL injection
- B) XSS
- C) CSRF
- D) Deadlock

 **Answer:** B

---

**71.**

`Filter order: Authorization → Action → Result`

Which runs first?

- A) Action
- B) Authorization
- C) Result
- D) Exception

 **Answer:** B

---

**72.**

```
Session[ "user " ]="A" ;
```

Session stored?

- A) Client
- B) Server
- C) View
- D) URL

 **Answer:** B

---

**73.**

```
await _service.GetDataAsync();
```

Thread usage?

- A) Blocks
- B) Frees thread
- C) Locks
- D) Sleeps

 **Answer:** B

---

**74. Which is NOT recommended?**

- A) Attribute routing
- B) Async actions
- C) Heavy logic in views
- D) AntiForgery token

 **Answer:** C

---

**75.**

```
[Authorize(Roles="Admin")]
```

Allows?

- A) All users
- B) Authenticated users

- C) Admin role only
- D) Anonymous

 **Answer:** C

---

### **76. Which prevents CSRF?**

- A) Html.Encode
- B) AntiForgeryToken
- C) Authorize
- D) HTTPS

 **Answer:** B

---

### **77. Which prevents XSS?**

- A) AntiForgeryToken
- B) Encoding output
- C) Authorize
- D) Routing

 **Answer:** B

---

### **78. Which is stateless?**

- A) Session
- B) Cookie
- C) MVC request
- D) Application

 **Answer:** C

---

### **79. Which breaks routing?**

- A) Wrong URL
- B) Missing controller
- C) Missing action
- D) All

 **Answer:** D

---

**80.**

```
app.UseStaticFiles();
```

Used for?

- A) Routing
- B) CSS/JS/images
- C) DB
- D) Security

 **Answer:** B

---

**81. Which improves load time?**

- A) Bundling
- B) Minification
- C) CDN
- D) All

 **Answer:** D

---

**82. Which executes on error?**

- A) Result filter
- B) Action filter
- C) Exception filter
- D) Auth filter

 **Answer:** C

---

**83. Which is most secure?**

- A) QueryString auth
- B) Forms authentication
- C) Hardcoded credentials
- D) ViewBag auth

 **Answer:** B

---

**84. Which causes 404?**

- A) Wrong route
- B) Missing action
- C) Missing controller
- D) All

 Answer: D

---

### 85. Which is MOST asked in CCEE?

- A) Bootstrap setup
- B) Routing + Security attributes
- C) Custom helpers syntax
- D) CSS files

 Answer: B

---

### 86. Which improves maintainability?

- A) Attribute routing
- B) Clean layout
- C) Filters
- D) All

 Answer: D

---

### 87. Which avoids thread blocking?

- A) Thread.Sleep
- B) Task.Delay
- C) Wait()
- D) Result

 Answer: B

---

### 88. Which ensures reusable UI?

- A) Layout
- B) Partial View
- C) Both
- D) None

 Answer: C

---

## **89. Which ensures authorization at action level?**

- A) Global filter
- B) [Authorize] attribute
- C) Route
- D) Layout

 **Answer:** B

---

# **SESSIONS 18 & 19 – ENTITY FRAMEWORK & RAZOR PAGES (CCEE MCQs)**

---

## **EASY LEVEL (20 MCQs)**

### **1. Entity Framework is:**

- A) UI framework
- B) ORM framework
- C) Logging framework
- D) Testing framework

 **Answer:** B

---

### **2. ORM stands for:**

- A) Object Relational Mapping
- B) Object Runtime Model
- C) Open Resource Model
- D) Object Reference Mapping

 **Answer:** A

---

### **3. EF is used to:**

- A) Write HTML
- B) Access database using objects
- C) Manage routing
- D) Handle sessions

 **Answer:** B

---

#### **4. Which namespace is commonly used for EF Core?**

- A) System.Data
- B) Microsoft.EntityFrameworkCore
- C) Microsoft.Data.SqlClient
- D) System.Linq

 **Answer:** B

---

#### **5. Which is NOT an EF approach?**

- A) Code First
- B) Database First
- C) Model First
- D) View First

 **Answer:** D

---

#### **6. Code First approach means:**

- A) DB created first
- B) Models created first
- C) Views created first
- D) Stored procedures first

 **Answer:** B

---

#### **7. Which attribute defines primary key?**

- A) [Required]
- B) [Key]
- C) [ForeignKey]
- D) [Index]

 **Answer:** B

---

## **8. Which attribute makes column mandatory?**

- A) [Key]
- B) [Required]
- C) [Range]
- D) [MaxLength]

 **Answer:** B

---

## **9. Which attribute defines relationship?**

- A) [Key]
- B) [ForeignKey]
- C) [Required]
- D) [NotMapped]

 **Answer:** B

---

## **10. Fluent API configuration is done in:**

- A) Controller
- B) Model class
- C) DbContext
- D) View

 **Answer:** C

---

## **11. Which class represents database session?**

- A) DbSet
- B) DbContext
- C) ModelBuilder
- D) Migration

 **Answer:** B

---

## **12. DbSet<T> represents:**

- A) Database
- B) Table

- C) Row
- D) Column

 **Answer:** B

---

### **13. Which command creates database?**

- A) Add-Migration
- B) Update-Database
- C) Scaffold-DbContext
- D) Drop-Database

 **Answer:** B

---

### **14. Migrations are used to:**

- A) Backup DB
- B) Version DB schema
- C) Secure DB
- D) Optimize queries

 **Answer:** B

---

### **15. Which method saves changes?**

- A) Save()
- B) Commit()
- C) SaveChanges()
- D) Execute()

 **Answer:** C

---

### **16. Razor Pages are:**

- A) MVC Controllers
- B) Page-based programming model
- C) ORM framework
- D) Middleware

 **Answer:** B

---

## **17. Razor Pages file extension is:**

- A) .cs
- B) .html
- C) .cshtml
- D) .razor

 **Answer:** C

---

## **18. Razor Pages logic resides in:**

- A) Controller
- B) PageModel
- C) ViewModel
- D) DbContext

 **Answer:** B

---

## **19. EF Core supports:**

- A) Only SQL Server
- B) Multiple databases
- C) Only Oracle
- D) No migrations

 **Answer:** B

---

## **20. Which is MOST asked in CCEE?**

- A) EF internals
- B) Code First + CRUD flow
- C) UI styling
- D) Razor syntax

 **Answer:** B

---

## **🟡 MEDIUM LEVEL (20 MCQs – WITH SNIPPETS)**

### **21.**

```
public class Emp {
```

[Key]

```
public int Id { get; set; }  
}
```

Id is:

- A) Foreign key
- B) Primary key
- C) Required field
- D) Index

 Answer: B

---

**22.**

```
public DbSet<Employee> Employees { get; set; }
```

Represents:

- A) Column
- B) Table
- C) Row
- D) Database

 Answer: B

---

**23.**

```
context.Employees.Add(emp);
```

Operation?

- A) Read
- B) Update
- C) Insert
- D) Delete

 Answer: C

---

**24.**

```
context.SaveChanges();
```

Effect?

- A) Commits changes
- B) Opens connection
- C) Closes connection
- D) Rolls back

 **Answer:** A

---

**25.**

[Required]

```
public string Name { get; set; }
```

Ensures?

- A) Primary key
- B) Non-null value
- C) Unique value
- D) Indexed value

 **Answer:** B

---

**26.**

```
[ForeignKey("Dept")]
```

```
public int DeptId { get; set; }
```

Defines?

- A) Primary key
- B) Navigation
- C) Foreign key
- D) Composite key

 **Answer:** C

---

**27.**

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
```

Used for?

- A) Routing
- B) Fluent API
- C) CRUD
- D) Validation

 **Answer:** B

---

**28.**

```
modelBuilder.Entity<Employee>()
    .HasKey(e => e.Id);
```

This is:

- A) Data Annotation
- B) Fluent API
- C) Validation
- D) Migration

 **Answer:** B

---

**29.**

```
Add-Migration Init
```

Does?

- A) Creates DB
- B) Creates migration file
- C) Updates DB
- D) Drops DB

 **Answer:** B

---

**30.**

```
Update-Database
```

Does?

- A) Creates models
- B) Applies migration
- C) Deletes data
- D) Creates controller

 **Answer:** B

---

**31.**

```
context.Employees.FirstOrDefault();
```

Returns?

- A) Collection
- B) Single entity or null
- C) Always exception
- D) Table

 **Answer:** B

---

**32.**

```
context.Employees.Find(1);
```

Finds by?

- A) Any column
- B) Primary key
- C) Foreign key
- D) Index

 **Answer:** B

---

**33.**

```
context.Employees.Remove(emp);
```

Operation?

- A) Insert
- B) Update
- C) Delete
- D) Select

 **Answer:** C

---

**34.**

[NotMapped]

```
public int Temp { get; set; }
```

Effect?

- A) Column created
- B) Ignored by EF
- C) Primary key
- D) Foreign key

 **Answer:** B

---

**35.**

```
public class AppDbContext : DbContext
```

Purpose?

- A) View logic
- B) DB connection & mapping
- C) Routing
- D) UI

 **Answer:** B

---

**36.**

```
options.UseSqlServer(cs);
```

Used for?

- A) Validation
- B) DB provider config
- C) Migration
- D) Security

 **Answer:** B

---

**37.**

```
public class IndexModel : PageModel
```

Used in?

- A) MVC
- B) Razor Pages
- C) Web API
- D) Middleware

 **Answer:** B

---

**38.**

```
public IActionResult OnGet()
```

Belongs to?

- A) Controller
- B) Razor Page
- C) Model
- D) DbContext

 **Answer:** B

---

**39.**

```
public IActionResult OnPost()
```

Triggered by?

- A) GET
- B) POST

- C) PUT
- D) DELETE

 **Answer:** B

---

#### **40. Which is TRUE?**

- A) Razor Pages replace MVC
- B) Razor Pages are page-centric
- C) Razor Pages need controllers
- D) Razor Pages cannot use EF

 **Answer:** B

---

### **HARD LEVEL (20 MCQs – TRICKY + SNIPPETS)**

#### **41.**

```
context.Employees.ToList();
```

Execution?

- A) Deferred
- B) Immediate
- C) Async
- D) Parallel

 **Answer:** B

---

#### **42.**

```
context.Employees.Where(e=>e.Id>5);
```

Execution?

- A) Immediate
- B) Deferred
- C) Cached
- D) Parallel

 **Answer:** B

---

**43.**

```
context.Employees.Single();
```

If more than one row?

- A) First returned
- B) null
- C) Exception
- D) Last returned

 **Answer:** C

---

**44.**

```
[MaxLength(10)]
```

Ensures?

- A) Min length
- B) Max length
- C) Exact length
- D) Regex

 **Answer:** B

---

**45.**

```
HasOne(e=>e.Dept).WithMany()
```

Defines?

- A) One-to-one
- B) One-to-many
- C) Many-to-many
- D) Inheritance

 **Answer:** B

---

**46.**

```
context.SaveChanges();
```

Returns?

- A) void
- B) int
- C) Task<int>
- D) Task

 **Answer:** C

---

**47.**

```
await context.SaveChanges();
```

Benefit?

- A) Blocks thread
- B) Frees thread
- C) Locks DB
- D) Speeds CPU

 **Answer:** B

---

**48.**

```
Add-Migration M1
```

```
Add-Migration M2
```

Then Update-Database applies?

- A) M2 only
- B) M1 only
- C) All pending migrations
- D) None

 **Answer:** C

---

**49.**

```
Scaffold-DbContext
```

Used for?

- A) Code First
- B) Database First
- C) Model First
- D) Migration

 **Answer:** B

---

**50.**

```
context.Employees.Update(emp);
```

If entity not tracked?

- A) Error
- B) Insert
- C) Update after attach
- D) Ignored

 **Answer:** C

---

**51. Which is NOT recommended?**

- A) Async EF calls
- B) DbContext per request
- C) Long-lived DbContext
- D) Migrations

 **Answer:** C

---

**52. Which enforces validation at DB level?**

- A) Razor validation
- B) Data annotations
- C) Fluent API
- D) Both B & C

 **Answer:** D

---

**53.**

## [ConcurrencyCheck]

Used for?

- A) Validation
- B) Concurrency control
- C) Indexing
- D) Security

 **Answer:** B

---

### 54.

#### UseLazyLoadingProxies()

Purpose?

- A) Eager loading
- B) Lazy loading
- C) Explicit loading
- D) No loading

 **Answer:** B

---

### 55. Which causes N+1 problem?

- A) Eager loading
- B) Lazy loading
- C) Explicit loading
- D) Joins

 **Answer:** B

---

### 56. Which avoids N+1 problem?

- A) Lazy loading
- B) Include()
- C) Navigation only
- D) Proxy

 **Answer:** B

---

**57.**

```
context.Employees.Include(e=>e.Dept)
```

Type?

- A) Lazy
- B) Eager loading
- C) Explicit loading
- D) Deferred

 **Answer:** B

---

**58. Which is MOST asked in CCEE?**

- A) Razor Pages internals
- B) EF Code First + CRUD
- C) EF source code
- D) SQL tuning

 **Answer:** B

---

**59. Which improves scalability?**

- A) Sync SaveChanges
- B) Async SaveChanges
- C) Thread.Sleep
- D) Locks

 **Answer:** B

---

**60. Which gives MAX score?**

- A) Memorizing commands
- B) Understanding EF flow + snippets
- C) UI styling
- D) SQL joins

 **Answer:** B

# SESSION 20 – Localization & Deployment (ASP.NET MVC Core – CCEE MCQs)

---

## EASY LEVEL (15 MCQs)

### 1. Localization is used to:

- A) Improve performance
- B) Support multiple languages/cultures
- C) Secure application
- D) Optimize database

 Answer: B

---

### 2. Localization mainly deals with:

- A) Routing
- B) Culture and language
- C) Authentication
- D) Logging

 Answer: B

---

### 3. Which namespace is commonly used for localization?

- A) System.Globalization
- B) System.Threading
- C) System.Resources
- D) Microsoft.AspNetCore.Localization

 Answer: D

---

### 4. Which object represents culture info?

- A) Culture
- B) CultureData

- C) CultureInfo
- D) Region

 **Answer:** C

---

## **5. Localization in ASP.NET Core is usually configured in:**

- A) appsettings.json
- B) Startup.cs / Program.cs
- C) web.config
- D) RouteConfig

 **Answer:** B

---

## **6. Which supports culture-specific formatting?**

- A) Thread
- B) ViewBag
- C) CultureInfo
- D) Session

 **Answer:** C

---

## **7. Resource files typically have extension:**

- A) .json
- B) .resx
- C) .xml
- D) .txt

 **Answer:** B

---

## **8. Deployment means:**

- A) Running app locally
- B) Publishing app to server
- C) Debugging code
- D) Writing unit tests

 **Answer:** B

---

## **9. ASP.NET Core applications can be deployed to:**

- A) IIS
- B) Azure
- C) Linux servers
- D) All of the above

 **Answer:** D

---

## **10. Which file stores environment-specific settings?**

- A) appsettings.json
- B) appsettings.Development.json
- C) web.config
- D) startup.xml

 **Answer:** B

---

## **11. Which command publishes ASP.NET Core app?**

- A) dotnet run
- B) dotnet build
- C) dotnet publish
- D) dotnet deploy

 **Answer:** C

---

## **12. Which server is commonly used with ASP.NET Core on Windows?**

- A) Apache
- B) Nginx
- C) IIS
- D) Tomcat

 **Answer:** C

---

## **13. Which environment is default if not specified?**

- A) Production
- B) Development
- C) Staging
- D) Test

 **Answer:** A

---

#### **14. Localization affects:**

- A) Data storage
- B) UI display
- C) Routing only
- D) Authentication

 **Answer:** B

---

#### **15. Which is MOST likely asked in CCEE?**

- A) Resource file syntax
- B) Localization concept + deployment basics
- C) Docker deployment
- D) Advanced globalization APIs

 **Answer:** B

---

### **MEDIUM LEVEL (15 MCQs – WITH SNIPPETS)**

#### **16.**

```
var culture = new CultureInfo("fr-FR");
```

This represents:

- A) French language
- B) French culture (France)
- C) Any European culture
- D) Neutral culture

 **Answer:** B

---

#### **17.**

```
Thread.CurrentThread.CurrentCulture = culture;
```

Effect?

- A) Changes UI language
- B) Changes number/date format
- C) Changes routing
- D) Changes authentication

 **Answer:** B

---

**18.**

```
services.AddLocalization();
```

Purpose?

- A) Enable routing
- B) Enable localization services
- C) Enable security
- D) Enable EF

 **Answer:** B

---

**19.**

```
app.UseRequestLocalization();
```

Used for?

- A) Logging
- B) Applying culture to requests
- C) Authentication
- D) Deployment

 **Answer:** B

---

**20.**

```
options.DefaultRequestCulture = new RequestCulture("en-US");
```

Meaning?

- A) Fallback culture
- B) Default culture

- C) UI culture
- D) Neutral culture

 **Answer:** B

---

**21.**

`Resources.Messages`

Represents?

- A) View
- B) Resource class
- C) Model
- D) Controller

 **Answer:** B

---

**22.**

`@Localizer[ "Welcome" ]`

Used for?

- A) Routing
- B) Localized text
- C) Validation
- D) Security

 **Answer:** B

---

**23. Which is TRUE?**

- A) Localization requires MVC only
- B) Localization works in Razor views
- C) Localization works only in controllers
- D) Localization works only in Web API

 **Answer:** B

---

**24.**

```
dotnet publish -c Release
```

This creates:

- A) Debug output
- B) Optimized build
- C) Source code
- D) Logs

 **Answer:** B

---

## 25. Which deployment model bundles .NET runtime?

- A) Framework-dependent
- B) Self-contained
- C) Portable
- D) Debug

 **Answer:** B

---

## 26. Which file configures IIS hosting?

- A) Program.cs
- B) Startup.cs
- C) web.config
- D) launchSettings.json

 **Answer:** C

---

## 27. Which environment enables detailed errors?

- A) Production
- B) Staging
- C) Development
- D) Test

 **Answer:** C

---

## 28.

```
ASPNETCORE_ENVIRONMENT=Production
```

Defines?

- A) Database
- B) Hosting environment
- C) Localization
- D) Security

 Answer: B

---

## 29. Which supports reverse proxy in deployment?

- A) IIS
- B) Kestrel
- C) Nginx
- D) All

 Answer: D

---

## 30. Which improves global user experience?

- A) Routing
- B) Localization
- C) Bundling
- D) Minification

 Answer: B

---

## HARD LEVEL (15 MCQs – TRICKY)

### 31.

`CultureInfo.CurrentCulture`

Controls?

- A) Number formatting
- B) Resource lookup
- C) Routing
- D) Authentication

 Answer: B

---

**32.**

`CultureInfo.CurrentCulture`

Controls?

- A) Resource strings
- B) UI language
- C) Date/number format
- D) Routing

 **Answer:** C

---

**33. Which is FALSE?**

- A) Localization supports culture fallback
- B) Resource files can be culture-specific
- C) Localization requires separate projects
- D) Localization works with Razor

 **Answer:** C

---

**34. Which causes wrong culture selection?**

- A) Missing resource file
- B) Incorrect culture code
- C) Wrong middleware order
- D) All

 **Answer:** D

---

**35.**

```
app.UseRequestLocalization();  
app.UseRouting();
```

Correct order?

- A) Yes
- B) No

 Answer: A

---

### 36. Which is NOT deployment concern?

- A) Connection string
- B) Environment variables
- C) Resource files
- D) LINQ queries

 Answer: D

---

### 37. Which is safest for production?

- A) Debug mode
- B) Development environment
- C) Release build
- D) Detailed errors

 Answer: C

---

### 38. Which enables zero-downtime deployment?

- A) Rebuild
- B) Restart IIS
- C) Blue-green deployment
- D) Debug publish

 Answer: C

---

### 39. Which breaks localization?

- A) Hardcoded strings
- B) Resource-based strings
- C) Culture settings
- D) Middleware

 Answer: A

---

### 40. Which is recommended?

- A) Hardcode UI text
- B) Use resource files
- C) Use ViewBag for text
- D) Store text in DB only

 **Answer:** B

---

#### **41. Which deployment option is cross-platform?**

- A) IIS only
- B) Azure App Service
- C) Linux + Nginx
- D) B and C

 **Answer:** D

---

#### **42. Which is MOST asked in CCEE?**

- A) Exact localization API
- B) Localization concept + deployment basics
- C) Azure DevOps pipelines
- D) Docker commands

 **Answer:** B

---

#### **43. Which improves maintainability?**

- A) Single-language UI
- B) Localization with resources
- C) Hardcoded text
- D) Duplicate views

 **Answer:** B

---

#### **44. Which is optional in CCEE exam?**

- A) Localization demo
- B) Deployment demo
- C) Concepts
- D) Configuration basics

 **Answer:** A & B (*Demo only topics*)

---

#### **45. Which gives easy marks in CCEE?**

- A) Remembering commands
- B) Understanding purpose of localization & deployment
- C) Writing full deployment scripts
- D) Advanced cloud concepts

 **Answer:** B

---

## **SESSIONS 21, 22 & 23 – WEB API, CORS & REACT INTEGRATION (CCEE MCQs)**

---

### **EASY LEVEL (20 MCQs)**

#### **1. Web API is mainly used to:**

- A) Render views
- B) Expose data over HTTP
- C) Manage sessions
- D) Handle routing only

 **Answer:** B

---

#### **2. Web APIs generally return data in:**

- A) HTML
- B) XML only
- C) JSON
- D) Razor

 **Answer:** C

---

#### **3. ASP.NET Core Web API follows:**

- A) MVC only
- B) REST principles
- C) SOAP
- D) WebForms

 Answer: B

---

#### 4. Which base class is commonly used for Web API controllers?

- A) Controller
- B) ApiController
- C) ControllerBase
- D) PageModel

 Answer: C

---

#### 5. Which attribute marks a Web API controller?

- A) [Controller]
- B) [ApiController]
- C) [WebApi]
- D) [Service]

 Answer: B

---

#### 6. Which HTTP verb retrieves data?

- A) POST
- B) PUT
- C) GET
- D) DELETE

 Answer: C

---

#### 7. Which HTTP verb creates new resource?

- A) GET
- B) POST
- C) PUT
- D) PATCH

 Answer: B

---

## **8. Which HTTP verb updates existing resource?**

- A) GET
- B) POST
- C) PUT
- D) TRACE

 **Answer:** C

---

## **9. Which HTTP verb deletes resource?**

- A) PUT
- B) DELETE
- C) GET
- D) POST

 **Answer:** B

---

## **10. CORS stands for:**

- A) Cross Origin Resource Sharing
- B) Common Object Resource System
- C) Client Origin Request Service
- D) Cross Object Request Security

 **Answer:** A

---

## **11. CORS issue occurs when:**

- A) Same origin request
- B) Browser blocks cross-origin request
- C) Server error
- D) DB error

 **Answer:** B

---

## **12. Which client commonly consumes Web APIs?**

- A) Razor View
- B) MVC Controller

- C) React App
- D) SQL Server

 **Answer:** C

---

### **13. Web API controllers usually return:**

- A) ViewResult
- B) IActionResult
- C) PartialView
- D) RazorPage

 **Answer:** B

---

### **14. Which library is used for JSON serialization?**

- A) System.Xml
- B) Newtonsoft.Json
- C) System.IO
- D) EntityFramework

 **Answer:** B

---

### **15. Which format does React mostly use to communicate?**

- A) XML
- B) SOAP
- C) JSON
- D) HTML

 **Answer:** C

---

### **16. Which is stateless?**

- A) MVC
- B) Web API
- C) Session
- D) TempData

 **Answer:** B

---

## **17. Which HTTP status means success?**

- A) 404
- B) 401
- C) 200
- D) 500

 **Answer:** C

---

## **18. Which status means resource created?**

- A) 200
- B) 201
- C) 400
- D) 500

 **Answer:** B

---

## **19. Which layer interacts with DB?**

- A) React
- B) Web API
- C) Browser
- D) Client JS

 **Answer:** B

---

## **20. Which topic is MOST important for CCEE?**

- A) React UI
- B) Web API flow + CORS
- C) CSS
- D) Bootstrap

 **Answer:** B

---

## **🟡 MEDIUM LEVEL (20 MCQs – WITH MANY SNIPPETS)**

### **21.**

[ApiController]

```
[Route("api/[controller]")]
public class ProductsController : ControllerBase {}
```

Purpose?

- A) MVC Controller
- B) Web API Controller
- C) Razor Page
- D) Middleware

 **Answer:** B

---

**22.**

```
[HttpGet]
```

```
public IActionResult Get() {}
```

Triggered by?

- A) POST
- B) PUT
- C) GET
- D) DELETE

 **Answer:** C

---

**23.**

```
[HttpPost]
```

```
public IActionResult Create(Product p) {}
```

p is populated using?

- A) ViewBag
- B) Session
- C) Model binding
- D) TempData

 **Answer:** C

---

**24.**

```
return Ok(products);
```

Returns HTTP?

- A) 404
- B) 500
- C) 200
- D) 201

 **Answer:** C

---

**25.**

```
return CreatedAtAction("Get", new {id=1}, obj);
```

Status code?

- A) 200
- B) 201
- C) 400
- D) 404

 **Answer:** B

---

**26.**

```
services.AddCors();
```

Purpose?

- A) Add routing
- B) Enable CORS services
- C) Add security
- D) Add EF

 **Answer:** B

---

**27.**

```
app.UseCors("AllowAll");
```

Effect?

- A) Disable security
- B) Apply CORS policy
- C) Enable routing
- D) Logging

 **Answer:** B

---

**28.**

```
builder.WithOrigins("http://localhost:3000");
```

Means?

- A) Block origin
- B) Allow specific origin
- C) Allow all origins
- D) Disable CORS

 **Answer:** B

---

**29.**

```
builder.AllowAnyHeader().AllowAnyMethod();
```

Effect?

- A) Restrict API
- B) Open CORS policy
- C) Error
- D) Block methods

 **Answer:** B

---

**30.**

```
[EnableCors("policy1")]
```

Applied to?

- A) Model
- B) Controller/Action
- C) View
- D) DbContext

 **Answer:** B

---

**31.**

```
HttpClient client = new HttpClient();
```

Used for?

- A) Hosting API
- B) Consuming API
- C) Routing
- D) DB access

 **Answer:** B

---

**32.**

```
var res = await client.GetAsync(url);
```

This is:

- A) Sync
- B) Async
- C) Parallel
- D) Blocking

 **Answer:** B

---

**33.**

```
JsonConvert.SerializeObject(obj);
```

Purpose?

- A) Deserialize JSON
- B) Convert object → JSON

- C) Save to DB
- D) Validate model

 **Answer:** B

---

**34.**

```
JsonConvert.DeserializeObject<Product>(json);
```

Purpose?

- A) Object → JSON
- B) JSON → Object
- C) Encryption
- D) Validation

 **Answer:** B

---

**35.**

```
fetch("https://localhost/api/products")
```

Called from?

- A) Server
- B) React/Browser
- C) DB
- D) API

 **Answer:** B

---

**36.**

```
fetch(url, { method:"POST", body:JSON.stringify(obj)})
```

Used for?

- A) GET
- B) UPDATE
- C) CREATE
- D) DELETE

 **Answer:** C

---

**37.**

[`HttpPut("{id}")`]

Used for?

- A) Create
- B) Read
- C) Update
- D) Delete

 **Answer:** C

---

**38.**

[`HttpDelete("{id}")`]

Used for?

- A) Read
- B) Create
- C) Update
- D) Delete

 **Answer:** D

---

**39.**

`return BadRequest();`

HTTP code?

- A) 200
- B) 400
- C) 401
- D) 500

 **Answer:** B

---

## 40. Which is TRUE?

- A) Web API returns views
- B) Web API is stateful
- C) Web API supports React
- D) Web API needs Razor

 Answer: C

---

## HARD LEVEL (20 MCQs – TRICKY + SNIPPETS)

### 41.

```
[HttpGet("{id}")]
public IActionResult Get(int id)
```

id comes from?

- A) Body
- B) Route
- C) Query only
- D) Session

 Answer: B

---

### 42.

```
public IActionResult Post([FromBody] Product p)
```

FromBody means?

- A) Route
- B) QueryString
- C) Request body
- D) Header

 Answer: C

---

### 43.

```
public IActionResult Get([FromQuery] int id)
```

id comes from?

- A) Route
- B) Body
- C) QueryString
- D) Header

 **Answer:** C

---

**44.**

[ ApiController ]

Automatically enables?

- A) Routing
- B) Model validation
- C) CORS
- D) EF

 **Answer:** B

---

**45.**

```
if(!ModelState.IsValid) return BadRequest(ModelState);
```

Purpose?

- A) Security
- B) Validation response
- C) Routing
- D) Logging

 **Answer:** B

---

**46.**

```
axios.post(url, data)
```

Used for?

- A) GET
- B) CREATE
- C) DELETE
- D) READ

 **Answer:** B

---

**47.**

```
services.AddControllers().AddNewtonsoftJson();
```

Purpose?

- A) Enable Razor
- B) JSON support
- C) Enable EF
- D) CORS

 **Answer:** B

---

**48.**

```
return StatusCode(500);
```

Meaning?

- A) Client error
- B) Server error
- C) Unauthorized
- D) Not found

 **Answer:** B

---

**49.**

```
fetch(url).then(res => res.json())
```

json() does?

- A) Serialize
- B) Deserialize

- C) Encrypt
- D) Validate

 **Answer:** B

---

## 50.

```
builder.AllowAnyOrigin();
```

Risk?

- A) SQL Injection
- B) XSS
- C) Security exposure
- D) Deadlock

 **Answer:** C

---

## 51. Which is **BEST** practice?

- A) AllowAnyOrigin in production
- B) Restrict origins
- C) Disable CORS
- D) Ignore security

 **Answer:** B

---

## 52.

```
await context.SaveChangesAsync();
```

Used in?

- A) React
- B) Web API
- C) Browser
- D) Client JS

 **Answer:** B

---

## 53. Which is end-to-end flow?

- A) React → MVC View
- B) React → Web API → DB
- C) Browser → DB
- D) MVC → Razor

 **Answer:** B

---

#### **54. Which status means unauthorized?**

- A) 400
- B) 401
- C) 404
- D) 500

 **Answer:** B

---

#### **55.**

[Authorize]

Effect?

- A) Allow all
- B) Restrict to authenticated users
- C) Disable API
- D) Enable CORS

 **Answer:** B

---

#### **56.**

OPTIONS request

Used for?

- A) CRUD
- B) CORS preflight
- C) Delete
- D) Update

 **Answer:** B

---

## **57. Which causes CORS error?**

- A) Same origin
- B) Missing headers
- C) Same port
- D) Same scheme

 **Answer:** B

---

## **58. Which is NOT RESTful?**

- A) GET /products
- B) POST /products
- C) GET /createProduct
- D) DELETE /products/1

 **Answer:** C

---

## **59. Which is MOST asked in CCEE?**

- A) React hooks
- B) HTTP verbs + CORS + flow
- C) Redux
- D) UI styling

 **Answer:** B

---

## **60. Which gives MAX marks?**

- A) Remembering React syntax
- B) Understanding Web API + CORS + snippets
- C) CSS
- D) Axios docs

 **Answer:** B

 **SESSIONS 24 & 25 – MVC INTEGRATION  
WITH REACT (CCEE MCQs)**

---

## EASY LEVEL (15 MCQs)

### 1. React is mainly used for:

- A) Database access
- B) Backend logic
- C) Frontend UI
- D) Authentication

 Answer: C

---

### 2. ASP.NET MVC is mainly used for:

- A) UI rendering only
- B) Backend logic & APIs
- C) Styling
- D) Browser state

 Answer: B

---

### 3. In MVC + React architecture, React acts as:

- A) Model
- B) Controller
- C) View
- D) Middleware

 Answer: C

---

### 4. React usually communicates with MVC using:

- A) Razor
- B) Session
- C) Web APIs
- D) TempData

 Answer: C

---

### 5. Data exchanged between React and MVC is usually in:

- A) XML
- B) HTML
- C) JSON
- D) CSV

 Answer: C

---

## 6. Which HTTP verb is used to fetch data?

- A) POST
- B) GET
- C) PUT
- D) DELETE

 Answer: B

---

## 7. Which HTTP verb is used to send new data?

- A) GET
- B) POST
- C) PUT
- D) PATCH

 Answer: B

---

## 8. Authentication in MVC + React is commonly handled using:

- A) ViewBag
- B) Session variables
- C) Tokens (JWT/Cookies)
- D) QueryString

 Answer: C

---

## 9. React applications run on:

- A) Server
- B) Database
- C) Browser
- D) IIS only

 Answer: C

---

## **10. Which is TRUE?**

- A) React replaces MVC completely
- B) MVC replaces React
- C) React and MVC work together
- D) React runs on server

 **Answer:** C

---

## **11. Which is responsible for routing in React?**

- A) MVC Routing
- B) React Router
- C) Web API
- D) DbContext

 **Answer:** B

---

## **12. Which is backend routing?**

- A) React Router
- B) MVC routing
- C) Browser routing
- D) Redux

 **Answer:** B

---

## **13. State in React refers to:**

- A) Database state
- B) UI data
- C) Session state
- D) Server memory

 **Answer:** B

---

## **14. Which improves UI responsiveness?**

- A) Server rendering only
- B) Client-side rendering

- C) Blocking calls
- D) Page reloads

 **Answer:** B

---

## **15. Which is MOST asked in CCEE?**

- A) React hooks
- B) MVC + React integration flow
- C) CSS frameworks
- D) JSX syntax

 **Answer:** B

---

## **MEDIUM LEVEL (15 MCQs)**

### **16. MVC + React integration is best done using:**

- A) Razor views only
- B) Web APIs
- C) TempData
- D) Partial views

 **Answer:** B

---

### **17. Which layer defines business logic?**

- A) React
- B) MVC Controller / Services
- C) Browser
- D) CSS

 **Answer:** B

---

### **18. Which handles state in React?**

- A) ViewBag
- B) useState / Redux
- C) Session
- D) DbContext

 **Answer:** B

---

## **19. Which is TRUE about React state?**

- A) Global by default
- B) Immutable
- C) Mutable directly
- D) Stored in DB

 **Answer:** B

---

## **20. React fetches data using:**

- A) SqlConnection
- B) HttpClient
- C) fetch / axios
- D) Razor helpers

 **Answer:** C

---

## **21. Authentication tokens are usually stored in:**

- A) ViewBag
- B) Session only
- C) Cookies / LocalStorage
- D) QueryString

 **Answer:** C

---

## **22. Which prevents unauthorized API access?**

- A) Routing
- B) [Authorize] attribute
- C) ViewModel
- D) CSS

 **Answer:** B

---

## **23. Which allows React app to call MVC APIs?**

- A) Routing
- B) Bundling

- C) CORS
- D) Layout

 **Answer:** C

---

#### **24. Which is recommended for large apps?**

- A) Monolithic React component
- B) Component-based structure
- C) Inline JS
- D) Razor scripts

 **Answer:** B

---

#### **25. Which manages application-wide state?**

- A) useState
- B) Redux / Context API
- C) ViewBag
- D) TempData

 **Answer:** B

---

#### **26. Which is TRUE?**

- A) React directly accesses DB
- B) MVC exposes APIs
- C) Browser calls DB
- D) Razor required

 **Answer:** B

---

#### **27. Which improves maintainability?**

- A) Hardcoded URLs
- B) Separation of concerns
- C) Mixing UI & logic
- D) Global variables

 **Answer:** B

---

## **28. Which is async communication?**

- A) Razor rendering
- B) fetch API calls
- C) Page reload
- D) Form post

 **Answer:** B

---

## **29. Which ensures secure communication?**

- A) HTTP
- B) HTTPS
- C) QueryString
- D) Cookies

 **Answer:** B

---

## **30. Which is best practice?**

- A) Logic in React only
- B) Logic in MVC services
- C) DB calls in React
- D) UI logic in controller

 **Answer:** B

---

## **HARD LEVEL (15 MCQs – TRICKY)**

### **31. In MVC + React, which is stateless?**

- A) React component
- B) Web API
- C) Session
- D) Cookie

 **Answer:** B

---

### **32. Which causes tight coupling?**

- A) API-based communication
- B) Shared DTOs

- C) Direct DB access from React
- D) JSON communication

 **Answer:** C

---

### **33. Which is TRUE?**

- A) React routing replaces MVC routing
- B) React routing handles frontend URLs
- C) MVC routing handles browser navigation
- D) Both routing are same

 **Answer:** B

---

### **34. Which is BEST authentication flow?**

- A) React → DB
- B) React → MVC → DB
- C) MVC → React
- D) Browser → DB

 **Answer:** B

---

### **35. Which avoids CORS issues?**

- A) Same origin
- B) AllowAnyOrigin in prod
- C) Disable security
- D) Hardcoding URLs

 **Answer:** A

---

### **36. Which should NOT be stored in React state?**

- A) UI flags
- B) Form values
- C) Sensitive secrets
- D) Component data

 **Answer:** C

---

### **37. Which improves scalability?**

- A) Sync API calls
- B) Async APIs
- C) Blocking UI
- D) Page reloads

 **Answer:** B

---

### **38. Which handles authorization?**

- A) React Router
- B) MVC Filters
- C) Redux
- D) View

 **Answer:** B

---

### **39. Which is correct responsibility split?**

- A) React – DB logic
- B) MVC – UI logic
- C) React – UI, MVC – API
- D) React – Routing only

 **Answer:** C

---

### **40. Which breaks security?**

- A) Token-based auth
- B) HTTPS
- C) Storing token in plain JS variable
- D) Authorization filters

 **Answer:** C

---

### **41. Which is recommended for React project structure?**

- A) Single file
- B) Feature-based folders
- C) Inline JSX everywhere
- D) No structure

 **Answer:** B

---

#### **42. Which is MOST tested in CCEE?**

- A) React hooks syntax
- B) Integration architecture
- C) JSX formatting
- D) CSS modules

 **Answer:** B

---

#### **43. Which improves testability?**

- A) Tight coupling
- B) API separation
- C) Direct DB calls
- D) Global state

 **Answer:** B

---

#### **44. Which is a best practice?**

- A) Hardcode API URLs
- B) Use environment configs
- C) Ignore errors
- D) Skip validation

 **Answer:** B

---

#### **45. Which gives MAX marks in CCEE?**

- A) Writing React UI
- B) Understanding MVC + React data flow
- C) Styling components
- D) JSX memorization

 **Answer:** B