1. Scenario: Handling Null References

Question: Consider the following C# code:

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

What will be the output of this code?

a) 0

b) NullReferenceException

c) 0, but only if the string is initialized

d) The code will not compile

Answer: b) NullReferenceException

---

2. Scenario: Foreach and Arrays

Question: Given the following C# code:

```csharp
int[] numbers = { 1, 2, 3, 4, 5 };
foreach (var number in numbers)
{
    number += 1;
}
Console.WriteLine(numbers[0]);
```

What will be printed on the console?

a) 1

b) 2

c) 0

d) The code will not compile

Answer: a) 1

---

3. Scenario: Structs and Reference Types

Question: Consider the following code:

```csharp
public struct Point
{
    public int X;
    public int Y;
}

Point p1 = new Point();
p1.X = 10;
Point p2 = p1;
p2.X = 20;

Console.WriteLine(p1.X);
```

What is the output of the above code?

a) 10

b) 20

c) 0

d) The code will not compile

Answer: a) 10

---

4. Scenario: Using Interfaces

Question: Given the following code:

```csharp
interface ITest
{
   void Print();
}

class Test : ITest
{
   public void Print()
   {
      Console.WriteLine("Hello from Test");
   }
}

Test obj = new Test();
obj.Print();
```

What will be the output?

a) Compile-time error

b) "Hello from Test"

c) No output, since the method is not called

d) "Test"

Answer: b) "Hello from Test"

---

5. Scenario: Exception Handling

Question: What will be the output of the following code?

```csharp
try
{
    int x = 0;
    int y = 10 / x;
}
catch (DivideByZeroException)
{
    Console.WriteLine("Divide by zero");
}
catch (Exception ex)
{
    Console.WriteLine("Some other exception: " + ex.Message);
}
finally
{
    Console.WriteLine("Finally block executed");
}
```

a) Divide by zero

b) Some other exception: Attempted to divide by zero

c) Finally block executed

d) Divide by zero

    Finally block executed

Answer: d) Divide by zero

Finally block executed

---

6. Scenario: LINQ Query

Question: What will be the output of the following code?

```csharp
int[] numbers = { 1, 2, 3, 4, 5 };
var result = from n in numbers
             where n % 2 == 0
             select n;

Console.WriteLine(result.Count());
```

a) 2

b) 3

c) 5

d) Compile-time error

Answer: a) 2

---

7. Scenario: Array Initialization

Question: Consider the following code:

```csharp
int[] arr = new int[5] { 1, 2, 3 };
Console.WriteLine(arr[4]);
```

What will be printed?

a) 0

b) 3

c) Compile-time error

d) IndexOutOfRangeException

Answer: a) 0

---

8. Scenario: Method Overloading

Question: What will be the output of the following code?

```csharp
void Display(int a)
{
    Console.WriteLine("Integer: " + a);
}

void Display(double a)
{
    Console.WriteLine("Double: " + a);
}

Display(5);
Display(5.5);
```

a) Integer: 5

b) Double: 5.5

c) Both a) and b)

d) Compile-time error

Answer: c) Both a) and b)

---

9. Scenario: Delegates

Question: What is the output of the following C# code?

```csharp
delegate void Del(string message);

static void Notify(string message)
{
    Console.WriteLine(message);
}

Del handler = Notify;
handler("Hello, Delegate");
```

a) Hello, Delegate

b) Compile-time error

c) NullReferenceException

d) Hello, Notify

Answer: a) Hello, Delegate

---

10. Scenario: String Immutability

Question: Given the following code:

```csharp
```

```
string str = "Hello";
str.ToUpper();
Console.WriteLine(str);
```

What will be the output?

a) HELLO

b) hello

c) Compile-time error

d) Hello

Answer: d) Hello

11. Scenario: Out Parameters

Question: What will be the output of the following code?

```csharp
void Calculate(out int x, out int y)
{
    x = 10;
    y = 20;
}

int a, b;
Calculate(out a, out b);
Console.WriteLine(a + ", " + b);
```

a) 0, 0

b) 10, 20

c) Compile-time error

d) 0, 20

Answer: b) 10, 20

---

12. Scenario: Static Constructors

Question: What will be the output of the following code?

```csharp
class MyClass
{
   static MyClass()
   {
      Console.WriteLine("Static constructor");
   }

   public MyClass()
   {
      Console.WriteLine("Instance constructor");
   }
}

MyClass obj1 = new MyClass();
MyClass obj2 = new MyClass();
```

a) Static constructor
     Instance constructor
     Instance constructor
b) Instance constructor
     Static constructor
     Instance constructor

c) Static constructor

Instance constructor

d) Instance constructor

Instance constructor

Answer: a) Static constructor

Instance constructor

Instance constructor

---

13. Scenario: Nullable Types

Question: What will be the output of the following code?

```csharp
int? x = null;
int y = x ?? -1;
Console.WriteLine(y);
```

a) null

b) 0

c) -1

d) Compile-time error

Answer: c) -1

---

14. Scenario: Method Hiding

Question: What will be the output of the following code?

```csharp
```

```
class BaseClass
{
    public void Show()
    {
        Console.WriteLine("BaseClass Show");
    }
}

class DerivedClass : BaseClass
{
    public new void Show()
    {
        Console.WriteLine("DerivedClass Show");
    }
}

BaseClass obj = new DerivedClass();
obj.Show();
```

a) BaseClass Show
b) DerivedClass Show
c) Compile-time error
d) BaseClass Show
DerivedClass Show

Answer: a) BaseClass Show

---

15. Scenario: Boxing and Unboxing

Question: What will be the output of the following code?

```csharp
int x = 5;
object obj = x;
x = 10;
Console.WriteLine((int)obj);
```

a) 5

b) 10

c) Compile-time error

d) InvalidCastException

Answer: a) 5

---

16. Scenario: Generics and Constraints

Question: What will happen when the following code is executed?

```csharp
class MyClass<T> where T : new()
{
    public T CreateInstance()
    {
        return new T();
    }
}

MyClass<string> obj = new MyClass<string>();
string instance = obj.CreateInstance();
```

a) A new instance of `string` will be created

b) Compile-time error

c) InvalidOperationException

d) NullReferenceException

Answer: b) Compile-time error

---

17. Scenario: Implicit and Explicit Conversion

Question: What will be the output of the following code?

```csharp
double d = 1234.7;
int i = (int)d;
Console.WriteLine(i);
```

a) 1234

b) 1235

c) 1234.7

d) Compile-time error

Answer: a) 1234

---

18. Scenario: Abstract Classes

Question: Given the following code:

```csharp
abstract class Animal
```

```
{
    public abstract void Sound();
}

class Dog : Animal
{
    public override void Sound()
    {
        Console.WriteLine("Bark");
    }
}

Animal myDog = new Dog();
myDog.Sound();
```

What will be the output?

a) Bark
b) Compile-time error
c) No output, since the method is abstract
d) NullReferenceException

Answer: a) Bark

---

19. Scenario: Enum Basics

Question: What will be the output of the following code?

```csharp
enum Days { Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday };
```

```
Console.WriteLine((int)Days.Wednesday);
```

a) 0

b) 3

c) 4

d) Compile-time error

Answer: b) 3

---

20. Scenario: Asynchronous Programming

Question: What will be the output of the following asynchronous code?

```csharp
async Task<int> CalculateAsync()
{
    await Task.Delay(1000);
    return 42;
}

Task<int> task = CalculateAsync();
Console.WriteLine(task.Result);
```

a) 0

b) 42

c) Compile-time error

d) InvalidOperationException

Answer: b) 42

21. Scenario: Property Getters and Setters

Question: What will be the output of the following code?

```csharp
class MyClass
{
    private int _value;

    public int Value
    {
        get { return _value; }
        set { _value = value * 2; }
    }
}

MyClass obj = new MyClass();
obj.Value = 5;
Console.WriteLine(obj.Value);
```

a) 5

b) 10

c) Compile-time error

d) 0

Answer: b) 10

---

22. Scenario: Inheritance and Constructors

Question: What will be the output of the following code?

```csharp
```

```
class BaseClass
{
   public BaseClass()
   {
      Console.WriteLine("BaseClass Constructor");
   }
}

class DerivedClass : BaseClass
{
   public DerivedClass()
   {
      Console.WriteLine("DerivedClass Constructor");
   }
}

DerivedClass obj = new DerivedClass();
```

a) BaseClass Constructor
    DerivedClass Constructor
b) DerivedClass Constructor
    BaseClass Constructor
c) Compile-time error
d) Only DerivedClass Constructor

Answer: a) BaseClass Constructor
    DerivedClass Constructor

---

23. Scenario: Indexers

Question: What will be the output of the following code?

```csharp
class MyCollection
{
    private int[] arr = new int[5];

    public int this[int index]
    {
        get { return arr[index]; }
        set { arr[index] = value; }
    }
}

MyCollection collection = new MyCollection();
collection[0] = 10;
collection[1] = 20;
Console.WriteLine(collection[1]);
```

a) 10

b) 20

c) 0

d) Compile-time error

Answer: b) 20

---

24. Scenario: Interface Inheritance

Question: What will be the output of the following code?

```csharp
interface IAnimal
{
    void Speak();
}

interface IDog : IAnimal
{
    void Bark();
}

class Dog : IDog
{
    public void Speak()
    {
        Console.WriteLine("Dog speaks");
    }

    public void Bark()
    {
        Console.WriteLine("Dog barks");
    }
}

IDog myDog = new Dog();
myDog.Bark();
```

a) Dog speaks

b) Dog barks

c) Compile-time error

d) No output

Answer: b) Dog barks

---

25. Scenario: Lambda Expressions

Question: What will be the output of the following code?

```csharp
Func<int, int> square = x => x * x;
Console.WriteLine(square(4));
```

a) 8

b) 16

c) 24

d) Compile-time error

Answer: b) 16

---

26. Scenario: Conditional Operator

Question: What will be the output of the following code?

```csharp
int x = 5;
string result = x > 10 ? "Greater" : "Lesser";
Console.WriteLine(result);
```

a) Greater

b) Lesser

c) 10

d) Compile-time error

Answer: b) Lesser

---

27. Scenario: Using 'is' Keyword

Question: What will be the output of the following code?

```csharp
object obj = "Hello";
if (obj is string)
{
    Console.WriteLine("It's a string");
}
else
{
    Console.WriteLine("It's not a string");
}
```

a) It's a string

b) It's not a string

c) Compile-time error

d) No output

Answer: a) It's a string

---

28. Scenario: Event Handling

Question: What will be the output of the following code?

```csharp
class MyEvent
{
    public event Action OnClick;

    public void Click()
    {
        if (OnClick != null)
            OnClick();
    }
}

MyEvent evt = new MyEvent();
evt.OnClick += () => Console.WriteLine("Button Clicked");
evt.Click();
```

a) Button Clicked

b) Compile-time error

c) No output

d) NullReferenceException

Answer: a) Button Clicked

---

29. Scenario: Exception Handling with Multiple Catch Blocks

Question: What will be the output of the following code?

```csharp
```

```
try
{
   int[] arr = new int[5];
   Console.WriteLine(arr[10]);
}
catch (IndexOutOfRangeException)
{
   Console.WriteLine("Index out of range");
}
catch (Exception)
{
   Console.WriteLine("General exception");
}
```

a) Index out of range

b) General exception

c) Compile-time error

d) No output

Answer: a) Index out of range

---

30. Scenario: Implicit and Explicit Interface Implementation

Question: What will be the output of the following code?

```csharp
interface ITest
{
   void Display();
}
```

```
class Test : ITest
{
    void ITest.Display()
    {
        Console.WriteLine("Display from ITest");
    }

    public void Display()
    {
        Console.WriteLine("Display from Test");
    }
}

Test obj = new Test();
obj.Display();
```

a) Display from ITest

b) Display from Test

c) Compile-time error

d) No output

Answer: b) Display from Test

31. Scenario: Access Modifiers

Question: What will be the output of the following code?

```csharp
class BaseClass
{
    protected int x = 10;
```

```
}

class DerivedClass : BaseClass
{
    public void PrintX()
    {
        Console.WriteLine(x);
    }
}

DerivedClass obj = new DerivedClass();
obj.PrintX();
```

a) 10
b) Compile-time error
c) 0
d) No output

Answer: a) 10

---

32. Scenario: Method Overriding
Question: What will be the output of the following code?
```csharp
class BaseClass
{
    public virtual void Display()
    {
        Console.WriteLine("BaseClass Display");
```

```
    }
}

class DerivedClass : BaseClass
{
    public override void Display()
    {
        Console.WriteLine("DerivedClass Display");
    }
}

BaseClass obj = new DerivedClass();
obj.Display();
```

a) BaseClass Display
b) DerivedClass Display
c) Compile-time error
d) No output

Answer: b) DerivedClass Display

---

33. Scenario: Delegate and Multicast
Question: What will be the output of the following code?
```csharp
delegate void MyDelegate();

class Program
{
```

```
    static void Method1()
    {
        Console.WriteLine("Method1");
    }

    static void Method2()
    {
        Console.WriteLine("Method2");
    }

    static void Main(string[] args)
    {
        MyDelegate del = Method1;
        del += Method2;
        del();
    }
}
```

a) Method1
b) Method2
c) Method1
    Method2
d) Compile-time error

Answer: c) Method1
    Method2

---

34. Scenario: Constructor Overloading

Question: What will be the output of the following code?

```csharp
class MyClass
{
   public MyClass()
   {
      Console.WriteLine("Default Constructor");
   }

   public MyClass(int x)
   {
      Console.WriteLine("Parameterized Constructor");
   }
}

MyClass obj1 = new MyClass();
MyClass obj2 = new MyClass(5);
```

a) Default Constructor
    Parameterized Constructor
b) Default Constructor
c) Parameterized Constructor
d) Compile-time error

Answer: a) Default Constructor
    Parameterized Constructor

---

35. Scenario: Type Inference with 'var'

Question: What will be the output of the following code?

```csharp
var x = 5;
var y = 10.5;
var result = x + y;
Console.WriteLine(result.GetType());
```

a) System.Int32

b) System.Double

c) System.String

d) Compile-time error

Answer: b) System.Double

---

36. Scenario: Extension Methods

Question: What will be the output of the following code?

```csharp
public static class Extensions
{
    public static void Print(this int value)
    {
        Console.WriteLine(value);
    }
}

int x = 5;
x.Print();
```

a) 5

b) Compile-time error

c) 0

d) No output

Answer: a) 5

---

37. Scenario: Implicit Conversion

Question: What will be the output of the following code?

```csharp
int x = 123456;
long y = x;
Console.WriteLine(y);
```

a) 123456

b) Compile-time error

c) OverflowException

d) 0

Answer: a) 123456

---

38. Scenario: Partial Classes

Question: What will be the output of the following code?

```csharp
partial class MyClass
```

```
{
    public void Method1()
    {
        Console.WriteLine("Method1");
    }
}

partial class MyClass
{
    public void Method2()
    {
        Console.WriteLine("Method2");
    }
}

MyClass obj = new MyClass();
obj.Method1();
obj.Method2();
```

a) Method1
b) Method2
c) Method1
Method2
d) Compile-time error

Answer: c) Method1
Method2

---

39. Scenario: Implicitly Typed Arrays

Question: What will be the output of the following code?

```csharp
var arr = new[] { 1, 2, 3.5 };
Console.WriteLine(arr.GetType());
```

a) System.Int32[]

b) System.Double[]

c) System.Object[]

d) Compile-time error

Answer: b) System.Double[]

---

40. Scenario: Custom Exceptions

Question: What will be the output of the following code?

```csharp
class MyException : Exception
{
    public MyException(string message) : base(message) {}
}

try
{
    throw new MyException("Custom exception");
}
catch (MyException ex)
{
    Console.WriteLine(ex.Message);
```

}
```

**a)** Custom exception

**b)** Compile-time error

**c)** Exception

**d)** No output

**Answer:** a) Custom exception

---

### 41. Scenario: Anonymous Types

**Question:** What will be the output of the following code?

```csharp
var obj = new { Name = "John", Age = 30 };
Console.WriteLine(obj.Name);
```

a) John

b) 30

c) Compile-time error

d) No output

Answer: a) John

---

42. Scenario: Tuples

Question: What will be the output of the following code?

```csharp

```
var tuple = (Name: "John", Age: 30);
Console.WriteLine(tuple.Age);
```

a) John

b) 30

c) Compile-time error

d) No output

Answer: b) 30

---

43. Scenario: Default Parameter Values

Question: What will be the output of the following code?

```csharp
void PrintMessage(string message = "Hello, World!")
{
    Console.WriteLine(message);
}

PrintMessage();
```

a) Hello, World!

b) Compile-time error

c) No output

d) NullReferenceException

Answer: a) Hello, World!

---

44. Scenario: Named Arguments

Question: What will be the output of the following code?

```csharp
void PrintDetails(string name, int age)
{
    Console.WriteLine($"Name: {name}, Age: {age}");
}

PrintDetails(age: 25, name: "Alice");
```

a) Name: Alice, Age: 25

b) Compile-time error

c) Name: 25, Age: Alice

d) NullReferenceException

Answer: a) Name: Alice, Age: 25

---

45. Scenario: Event with Multiple Handlers

Question: What will be the output of the following code?

```csharp
class MyEvent
{
    public event Action OnClick;

    public void Click()
    {
```

```
        OnClick?.Invoke();
    }
}

MyEvent evt = new MyEvent();
evt.OnClick += () => Console.WriteLine("Handler 1");
evt.OnClick += () => Console.WriteLine("Handler 2");
evt.Click();
```

a) Handler 1
b) Handler 2
c) Handler 1
    Handler 2
d) Compile-time error

Answer: c) Handler 1
    Handler 2

---

46. Scenario: Using the `using` Statement
Question: What will be the output of the following code?
```csharp
using (var sw = new System.IO.StringWriter())
{
    sw.Write("Hello");
    Console.WriteLine(sw.ToString());
}
```

a) Hello

b) Compile-time error

c) No output

d) NullReferenceException

Answer: a) Hello

---

47. Scenario: LINQ with Objects

Question: What will be the output of the following code?

```csharp
int[] numbers = { 1, 2, 3, 4, 5 };
var result = numbers.Where(n => n % 2 == 0);
Console.WriteLine(result.Count());
```

a) 2

b) 3

c) 5

d) Compile-time error

Answer: a) 2

---

48. Scenario: Abstract Method Implementation

Question: What will be the output of the following code?

```csharp
abstract class Animal
{
```

```
    public abstract void MakeSound();
}

class Dog : Animal
{
    public override void MakeSound()
    {
        Console.WriteLine("Bark");
    }
}

Animal myDog = new Dog();
myDog.MakeSound();
```

a) Bark
b) Compile-time error
c) No output
d) NullReferenceException

Answer: a) Bark

---

49. Scenario: `params` Keyword
Question: What will be the output of the following code?
```csharp
void PrintNumbers(params int[] numbers)
{
    foreach (var number in numbers)
    {
```

```
        Console.WriteLine(number);
    }
}

PrintNumbers(1, 2,

 3, 4);
```

a) 1 2 3 4

b) Compile-time error

c) No output

d) 1234

Answer: a) 1 2 3 4

---

50. Scenario: Handling Null with `?.`

Question: What will be the output of the following code?

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

a) 0

b) null

c) Compile-time error

d) NullReferenceException

Answer: b) null