

1. **Hello World Comparison:** Write a simple "Hello World" program in C#, outputting the string to the Unity console during the Awake phase of a GameObject's instantiation.

using UnityEngine;

public class HelloWorld : MonoBehaviour

```
{  
    private void Awake()  
    {  
        Debug.Log("Hello World");  
    }  
}
```

2. **Matching: Java to C# Keywords** Match the Java keyword with its C# equivalent. Write down the correct sequence of letters:

Java
extends
final (for variables)
super (in constructors)
@Override

C#
A. override
B. base
C. readonly
D. :

Sequence: **DCBA**

3. **Constructors:** Translate this Java constructor to C#:

```
public ClassName(String name) {  
    super(name);  
}
```

```
public ClassName(string name) : base(name)  
{  
}
```

4. **Autoimplemented Fields:** What is this? Choose all that apply.

```
public string Name { get; set; }
```

- a) A property with a backing field created by the compiler
- b) A property that would be utilized with myInstance.Name()
- c) An auto-implemented property
- d) A property that becomes immutable after the constructor is run

A,C

- 5. Overriding:** Given the following C# code, the intention is for the Child class to override the DisplayMessage method of the Parent class. However, the code is not currently set up correctly for overriding. Modify the code to correctly implement overriding.

```
public class Parent
{
    public void DisplayMessage()
    {
        Debug.Log("Message from Parent class");
    }
}

public class Child : Parent
{
    public void DisplayMessage()
    {
        Debug.Log("Message from Child class");
    }
}
```

```
public class Parent
{
    public virtual void DisplayMessage()
    {
        Console.WriteLine("Message from Parent class");
    }
}
```

```
public class Child : Parent
{
    public override void DisplayMessage()
    {
        Console.WriteLine("Message from Child class");
    }
}
```

- 6. Reference vs Value Types:** When should you use the `ref` keyword in C#?
- a) When declaring static methods
 - b) When you want to pass a parameter by value
 - c) When you want to pass a parameter by reference
 - d) When overriding methods

C

7. **Structs:** Define a simple C# struct for a color with `red`, `green`, and `blue` values.

// used bytes to optimize storage

```
public struct Color
{
    public byte Red { get; }
    public byte Green { get; }
    public byte Blue { get; }

    public Color(byte red, byte green, byte blue)
    {
        Red = red;
        Green = green;
        Blue = blue;
    }
}
```

8. **Nullable Types:** Which of the following is a nullable int declaration in C#?

- a) `int number;`
- b) `int number = null;`
- c) `int? number;`
- d) `?int number;`

C

9. **Null Operators:** Translate the following Java code to C# using the null conditional operator:

```
if (object != null) {
    object.method();
}
```

`object?.method();`

10. **Permissions:** If a Java class field is `protected`, what would be its most similar accessibility in C#?

- a) `private`
- b) `public`
- c) `internal`
- d) `protected`

D, the `protected` keyword exists in C#

11. **Multidimensional Arrays:** Which of these is a 2x2 multidimensional array of integers in C#? Choose all that apply.

- a) `int[][] array = new int[2][2];`
- b) `int[,] array = new int[2, 2];`
- c) `int[2,2] array = new int[2,2];`
- d) `int[2][2] array = new int[2][2];`

B

12. **Java's final in C#:** Which C# keyword is equivalent to Java's `final` when declaring a variable that must not change its value?

- a) `static`
- b) `constant`
- c) `readonly`
- d) `final`

C

13. **instanceof in C#:** Rewrite the following Java code in C#. Please observe all C# norms:

```
if (object instanceof String) {  
    otherMMethod();  
}
```

```
if (obj is string)  
{  
    otherMethod();  
}
```

14. **Null Coalescing:** Use the null coalescing operator to rewrite this line of C# code:

```
string name = (possibleName != null) ? possibleName : "Default";
```

```
string name = possibleName ?? "Default";
```

15. **Integrated Extra Question:** Given a potential null object `name`, use the null conditional operator to call a method `Print()` on it and use the null coalescing operator to provide "Not found" as a default value if `name` is null.

```
string result = name?.Print() ?? "Not found";
```