In C#, **string interpolation** by default uses the **current culture** for formatting numbers.
If you want to ensure **InvariantCulture** (culture-independent formatting, e.g., decimal separator is always `"."`), you need to explicitly format the values.

✅ Example: Interpolated String with `InvariantCulture`

```
using System;
using System.Globalization;

class Program
{
    static void Main()
    {
        double number = 12345.6789;

        // Interpolated string with InvariantCulture
        string result = string.Format(
            CultureInfo.InvariantCulture,
            $"The number is {number:F2}"
        );

        Console.WriteLine(result); // Output: The number is 12345.68
    }
}
```

🔹 Alternative: Format Each Value Explicitly

```
using System;
using System.Globalization;

class Program
{
    static void Main()
    {
        double number = 12345.6789;
```

```
        // Explicit formatting inside interpolation
        string result = $"The number is {number.ToString("F2",
CultureInfo.InvariantCulture)}";

        Console.WriteLine(result); // Output: The number is 12345.68
    }
}
```

🔹 .NET 8+ Shortcut: `$""` with `InvariantCulture`

If you are using **.NET 8 or later**, you can use the new **`Invariant` interpolated string handler**:

```
using System;
using System.Globalization;

class Program
{
    static void Main()
    {
        double number = 12345.6789;

        string result = FormattableString.Invariant($"The number is
{number:F2}");
        Console.WriteLine(result); // Output: The number is 12345.68
    }
}
```

Summary

- **Pre-.NET 8** → Use **`string.Format(CultureInfo.InvariantCulture, ...)`** or **`ToString(..., InvariantCulture)`**.
- **.NET 8+** → Use **`FormattableString.Invariant(...)`** for cleaner syntax.

If you want, I can also show you **how to globally force all interpolated strings in your project to use `InvariantCulture`** without repeating the formatting code.

Do you want me to prepare that?