

# CsharpDay03Assignment

Name: Mohamed Waleed Elkady

## PART01

- 1) Question: What is the difference between `int.Parse` and `Convert.ToInt32` when handling null inputs?

*int.Parse throws an ArgumentNullException when the input is null, while Convert.ToInt32 returns 0 when the input is null.*

- 2) Question: Why is TryParse recommended over Parse in user-facing applications?

*TryParse is recommended because it does not throw exceptions for invalid input, improves performance, prevents application crashes,*

*and safely handles user input by returning a boolean result.*

- 3) Question: Explain the real purpose of the GetHashCode() method.

*GetHashCode generates a numeric value used to identify objects in hash-based collections like Dictionary and HashSet, allowing faster searching and comparison.*

- 4) What is the significance of reference equality in .NET?

*Reference equality means two variables refer to the same object in memory, so changes made through one reference affect the other, which is essential for understanding object behavior and memory management.*

- 5) Question: Why is string immutable in C#?

*String is immutable to ensure security, thread safety, efficient memory usage through string pooling, and predictable behavior since any modification creates a new string object.*

- 6) Question: How does StringBuilder address the inefficiencies of string concatenation?

*StringBuilder modifies text in the same memory location without creating new objects, reducing memory allocation and garbage collection overhead.*

- 7) Question: Why is StringBuilder faster for large-scale string modifications?

*StringBuilder uses a dynamic internal buffer that grows as needed, minimizing memory*

*allocations and making it efficient for frequent or large text changes.*

- 8) Question: Which string formatting method is most used and why?

*String interpolation is the most used because it is easy to read, concise, less error-prone, and closely resembles natural language formatting.*

- 9) Question: Explain how StringBuilder is designed to handle frequent modifications compared to strings.

*StringBuilder is mutable and allows in-place changes using methods like Append, Insert, Replace, and Remove, making it far more efficient than strings which create new objects on every modification.*

## PART02



Mohamed Waleed Elkady • You

Undergraduate Electronics and Communication Engineer / Full stack .Net dev...

1m • ⑤

...

لية الـ String في C# ... والفكرة اللي مش كل مبرمج فاهمها 😊

ناس كتير بتعامل مع الـ string في C على إنه نوع بيانات عادي، تكتب فيه وتعلّل عليه وخلاص، بس الحقيقة إن كون الـ string immutable مش تفصيلة تقنية صغيرة، ده قرار تصميم مؤثر على الأمان، الأداء، وطريقة تفكيرك كمبرمج. خلينا ننص عليها من زاوية مختلفة.

تخيل إن عندك string مستخدم ك Token أو Connection String، واتبعث أكثر من method في البرنامج، لو الـ string كان mutable، أي تعديل سبّي في مكان واحد ممكن يتوّظ سلوك البرنامج كله في أماكن تانية من غير ما تحس. الـ immutability هنا مش رفاهية، دي حماية. النقطة الثانية اللي ناس كتير بتعديها هي إن الـ immutability بتخلّي الـ string Thread-safe تلقائياً.

يعني إيه؟

يعني لو عندك أكثر من thread بيقرأ نفس الـ string، مفيش خوف إن واحد يغيره والتاني يقرأ قيمة نص متعدلة نص لأنّ ده بيوفّر مجهود ضخم كان المفترض بتعمل في synchronization.

طب والأداء؟

هنا المفارقة اللي يتلّبّط ناس كتير.

آه، تعديل string بيعمل object جديد، وده شكله مكلف.

لكن في المقابل، C# بيستخدم String Interning، وده معناه إن strings المتكررة بتتشارع على نفس المكان في الذاكرة بدل ما تتكرّر.

يعني immutability نفسها هي اللي سمحّت بالتحسين ده.

بس هل ده معناه إن string دائمًا اختيار صح؟

طبعاً لأنّ

لو بتبني نص كبير خطوة خطوة، خصوصاً جوه loops، استخدام string هنا هو anti-pattern في الحالة دي، StringBuilder مش مجرد اختيار أسرع، ده اختيار صح من الأساس.

الخلاصة؟

الفرق بين developer junior و developer senior فاهم بعد مش إنه يعرف إن الـ string immutable

لكن إنه يعرف إنّي ده مفید... وإنّي يكون خطر على الأداء.

اللغة مش بس syntax،

اللغة فلسفة وتصميم... والـ string في C #مثال واضح على ده 😊

CSharp #DotNet #Programming #SoftwareDevelopment #CodingTips#

#TechInEgypt #StringImmutability #DevLife #CodeSmart #LearnCSharp

#SoftwareEngineering #ProgrammingTips #ThreadSafety #MemoryManagement

#StringBuilder



2) What is Enum data type, when is it used, and name three common built-in enums used frequently?

*Enum (enumeration) is a value type in C# used to define a set of named constant values, making code more readable, maintainable, and type-safe. It is used when a variable should only have a limited number of predefined values, such as days, states, or options. Three commonly used built-in enums are DayOfWeek, ConsoleColor, and Environment.SpecialFolder.*

3) What are the scenarios to use string vs StringBuilder?

*String should be used when the text value does not change frequently, such as constants, display messages, or simple concatenations, because strings are immutable. StringBuilder should be used when the text changes frequently, such as in loops, large text processing, or dynamic content generation, because it is mutable and more memory-efficient.*

