**1. What is the difference between int.Parse and Convert.ToInt32 when handling null inputs?**

* int.Parse(null) throws an ArgumentNullException.
* Convert.ToInt32(null) returns 0.

**2. Why is TryParse recommended over Parse in user-facing applications?**

* TryParse prevents exceptions by returning a boolean indicating success or failure, making it safer and more user-friendly.

**3. Explain the real purpose of the GetHashCode() method.**

* It returns a numeric value used in hashing algorithms, especially for quick lookup in collections like Dictionary and HashSet.

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

* It checks if two references point to the **same object in memory**, not just equal content.

**5. Why is string immutable in C#?**

* To ensure thread safety, reduce memory issues, and enable string interning for performance optimization.

**6. How does StringBuilder address the inefficiencies of string concatenation?**

* It modifies a **single mutable buffer** instead of creating a new string on each operation.

**7. Why is StringBuilder faster for large-scale string modifications?**

* Because it avoids the repeated memory allocation and copying that immutable strings require during concatenation.

**8. Which string formatting method is most used and why?**

* **String interpolation ($"...")** is most used for its **readability**, **clarity**, and **inline variable support**.

**9. Explain how StringBuilder is designed to handle frequent modifications compared to strings.**

* It maintains an internal buffer that can grow dynamically, enabling fast and memory-efficient edits without creating new objects each time.