

1. **Purpose of the finally block:**

Runs code regardless of whether an exception occurs, typically for cleanup (e.g., closing files, releasing resources).

2. **How `int.TryParse()` improves robustness compared to `int.Parse()`:**

`TryParse()` returns false instead of throwing an exception if parsing fails, preventing crashes and avoiding try-catch overhead.

3. **Exception when accessing `.Value` on a null `Nullable<T>`:**

`InvalidOperationException`.

4. **Why check array bounds before accessing elements:**

Prevents `IndexOutOfRangeException` and avoids reading/writing invalid memory.

5. **How `GetLength(dimension)` is used:**

Returns the number of elements in a specific dimension of a multi-dimensional array (0 for rows, 1 for columns).

6. **Memory allocation difference (jagged vs rectangular arrays):**

- a. **Rectangular array:** single continuous memory block for all elements.
- b. **Jagged array:** array of separate arrays, each row stored independently in memory.

7. **Purpose of nullable reference types in C#:**

Makes nullability explicit in the type system, enabling compiler warnings for possible null dereferences.

8. Performance impact of boxing/unboxing:

- a. **Boxing:** Allocates a new object on the heap (extra memory + time).
- b. **Unboxing:** Requires type check and value copy, slower than direct value type operations.

9. Why must out parameters be initialized inside the method:

C# requires all out parameters to be assigned before the method exits to ensure the caller gets a definite value.

10. Why optional parameters must appear at the end:

Ensures that positional arguments match parameters correctly and avoids ambiguity in method calls.

11. How null propagation operator (?.) prevents NullReferenceException:

Stops evaluation and returns null immediately if the object before ?. is null.

12. When a switch expression is preferred over if statements:

- a. When mapping one value to another concisely.
- b. Improves readability for multiple fixed comparisons.

13. Limitations of the params keyword:

- a. Only one params parameter per method.
- b. Must be the last parameter.
- c. Works only with a single-dimensional array.