Assignment DAY 05

1) What is the purpose of the finally block?

The finally block ensures that specific code runs regardless of whether an exception occurs. It's commonly used for cleanup actions like closing files or releasing resources.

2) How does int.TryParse() improve program robustness compared to int.Parse()?

int.TryParse() avoids exceptions by returning a boolean indicating success or failure, instead of throwing an exception if the input is invalid. This prevents runtime crashes from bad input.

3) What exception occurs when trying to access Value on a null Nullable<T>?

InvalidOperationException occurs when attempting to access the Value property of a null Nullable<T>.

4) Why is it necessary to check array bounds before accessing elements?

Accessing out-of-bound indices throws IndexOutOfRangeException. Checking bounds prevents such errors and ensures program stability.

5) How is the GetLength(dimension) method used in multi-dimensional arrays?

GetLength(dimension) returns the size of the specified dimension in a multi-dimensional array. For example, array.GetLength(0) gives the number of rows.

6) How does the memory allocation differ between jagged arrays and rectangular arrays?

Rectangular arrays allocate memory in a single block for all elements. Jagged arrays allocate memory separately for each row, allowing different row sizes and saving space.

7) What is the purpose of nullable reference types in C#?

Nullable reference types help detect and avoid null-related bugs at compile time by enabling the compiler to warn when a variable might be null.

8) What is the performance impact of boxing and unboxing in C#?

Boxing and unboxing introduce overhead because they involve converting between value types and reference types, resulting in additional memory allocation and processing.

9) Why must out parameters be initialized inside the method?

The method is responsible for assigning values to out parameters before the method ends, ensuring the caller receives valid output.

10) Why must optional parameters always appear at the end of a method's parameter list?

Optional parameters must come last to avoid ambiguity during method calls and ensure proper matching of arguments.

11) How does the null propagation operator prevent NullReferenceException?

It short-circuits property or method access if the preceding object is null, returning null instead of throwing an exception.

12) When is a switch expression preferred over a traditional if statement?

Switch expressions are concise and readable for multiple constant-based conditions. They reduce code size and improve clarity.

13) What are the limitations of the params keyword in method definitions?

Only one params parameter is allowed, and it must be the last in the method signature. It cannot be used with ref or out.