Part01:

Question01:

The finally block is used to execute a set of statements regardless of whether an exception is thrown within the associated try or catch blocks. Its primary purpose is to ensure essential cleanup actions, such as releasing system resources.

Question02:

int.TryParse() improves program robustness compared to int.Parse() primarily by handling invalid input without throwing exceptions.

Question03:

When you try to access the Value property on a Nullable<T> instance that is null a System.InvalidOperationException is thrown.

Question04:

It is necessary to check array bounds before accessing elements because failing to do so results in undefined behavior, which can lead to unpredictable program crashes, data corruption, and serious security vulnerabilities.

Question05:

The GetLength(dimension) method is used to retrieve the number of elements in a specific dimension of a multi-dimensional array.

Question06:

Jagged arrays, which are arrays of arrays, allocate memory for each inner array individually, allowing rows to have different lengths and reducing memory waste for non-rectangular data. In contrast, rectangular array allocate a single contiguous block of memory for all elements, which is faster to access but requires fixed row/column sizes.

Question07:

The purpose of nullable reference type in C# is to minimize the likelihood of System.NullReferenceException runtime errors by providing compile-time warnings and enabling developers to explicitly declare their intent regarding whether a reference can be null.

Question08:

Boxing and unboxing in C# are computationally expensive operations that can significantly impact application performance, especially in performance-critical code or frequent operations. The overhead stems from memory management and CPU processing costs.

Question09:

out parameters must be assigned a value inside the method before it returns, as a compile-time guarantee that the variable will not hold an undefined value in the calling code.

Question10:

Optional parameters must appear at the end of a method's parameter list to avoid ambiguity in argument matching when the method is called using positional arguments.

Question11:

The null conditional operator (?.) prevents a NullReferenceException by performing an implicit null check before attempting to access a member (property, method, or indexer) on an object. This mechanism is called short-circuiting.

Question12:

A switch expression is preferred over a traditional if statement when you are evaluating a single variable or expression against multiple, discrete constant values, particularly when the goal is to return a value based on that match.

Question13:

The params keyword in C# is a powerful tool for creating flexible methods that accept a variable number of arguments, but it comes with strict compiler constraints to ensure code clarity and prevent ambiguity.