Task report

Course: CSharp

Day: 03

Name: bothina marwan naeem

Group: GIZ4_SWD5_S3

Code: 21021252

Questions

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

- int.parse(null) -> throwsArgumentNullExption
- convert.toint32(null) -> returns zero
- convert.toint32 is more secure with null

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

- no exceptions
- faster performance
- suitable for direct user input handling
- prevent the program from crashing

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

- It is used to determine the position of an object in hash-based collections such as Dictionary and Hashtable.
- It helps improve performance when searching for objects.
- It is not a unique identifier for an object.

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

Reference equality means multiple references point to the same object in memory,

so changes through one reference affect all others.

5. Why string is immutable in C#?

- It is thread-safe.
- It improves performance through the String Pool mechanism.
- It prevents unexpected changes to string values.
- Any modification creates a new object in memory instead of changing the original one.

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

- Modifies the same object in memory
- Avoids creating new string objects on each change
- Reduces memory allocation
- Minimizes garbage collection overhead
- Improves performance, especially with frequent modifications

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

- It modifies the same object instead of creating new ones
- It uses less memory allocation
- It reduces Garbage Collection overhead
- It is optimized for frequent and large string changes
- It provides better overall performance

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

- String Interpolation ($) is the most commonly used method.
- It is easy to read and write.
- It is less error-prone compared to concatenation and string.Format.
- It makes the code cleaner and more maintainable.

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

- StringBuilder is mutable, while strings are immutable.
- It modifies the same object in memory instead of creating new objects.
- It uses an internal buffer that can grow as needed.
- It reduces memory allocation and garbage collection.
- It provides better performance for frequent or large string modifications.

Part2

- What's Enum data type, when is it used? And name three common built_in enums used frequently?

- Enum (Enumeration) is a value data type used to define a set of named constant values.

It makes the code more readable, organized, and type-safe.

When is Enum used?

- When a variable has a fixed set of possible values
- To replace magic numbers
- To improve code clarity and maintainability

Common built-in enums:

- DayOfWeek
- ConsoleColor
- Environment.SpecialFolder

-what are scenarios to use string Vs StringBuilder?

Use string when:

- The value does not change frequently
- Working with small or simple text
- Readability is more important than performance

Use StringBuilder when:

- Performing frequent modifications (append, insert, replace)
- Working with large strings
- Using loops for string manipulation
- Performance and memory efficiency are important

Part3

- what meant by user defined constructor and its role in initialization?

A user-defined constructor is a constructor that is explicitly created by the programmer inside a class.

Role in initialization:

- It is used to initialize object data members at the time of object creation.
- It allows passing parameters to set initial values.
- It ensures the object starts in a valid and consistent state.
- It gives the programmer control over initialization logic.

- **compare between Array and Linked List**

Array:

- Stores elements in contiguous memory locations
- Has a fixed size
- Provides fast access using index
- Insertion and deletion are costly due to shifting elements
- Uses less memory overhead

Linked List:

- Stores elements in non-contiguous memory locations
- Has a dynamic size
- Allows sequential access only
- Insertion and deletion are easier and faster
- Uses more memory because it stores references