**Assignment-1**

Test your knowledge:

1. What type would you choose for the following “numbers”?

A person’s telephone number - Type: **string**

A person’s height - Type: **float or decimal**

A person’s age - Type: **int**

A person’s gender (Male, Female, Prefer Not to Answer) - Type: **enum or string**

A person’s salary - Type: **decimal**

A book’s ISBN - Type: **string**

A book’s price - Type: **decimal**

A book’s shipping weight - Type: **float or decimal**

A country’s population - Type: **long**

The number of stars in the universe - Type: **long**

The number of employees in each of the small or medium businesses in the United Kingdom (up to about 50,000 employees per business) - Type: **int**

2. What are the differences between value type and reference type variables? What is boxing and unboxing?

* **Value Types:** they directly contain their data. Each variable has its own copy of the data. They are stored on the stack.
* Examples: int, float, char
* user defined value types of examples: struct, enum.
* **Reference Types:** Reference types contain a reference to their data. Multiple variables can reference the same object. They are stored on the heap.
* Examples: class, string, array, delegate.
* Boxing is the process of converting a value type to a reference type. When a value type is boxed, it is wrapped inside an object and stored on the heap. Whereas, Unboxing is the process of converting a reference type back to a value type. When an object that contains a boxed value type is unboxed, the value is extracted from the object.

3. What is meant by the terms managed resource and unmanaged resource in .NET

**Managed Resource:** Managed resources are resources that are managed by the .NET runtime's garbage collector. These include objects created using .NET classes, such as instances of types like string, ArrayList, and custom objects.

**Unmanaged Resource:** Unmanaged resources are resources that are not handled by the .NET garbage collector and must be managed manually. These include resources like file handles, database connections, and memory allocated using unmanaged code.

4. Whats the purpose of Garbage Collector in .NET?

The Garbage Collector automatically manages the allocation and release of memory in .NET applications. It ensures that objects that are no longer needed by the application are properly disposed of, freeing up memory resources. By handling memory management automatically, the Garbage collector helps prevent memory leaks and other related issues, allowing developers to focus more on application logic rather than memory management.

1. What happens when you divide an int variable by 0?

Answer: Throws System.DivideByZeroException.

2. What happens when you divide a double variable by 0?

Answer: Results in Infinity, -Infinity, or NaN.

3. What happens when you overflow an int variable, that is, set it to a value beyond its range?

Answer: Wraps around in an unchecked context; throws System.OverflowException in a checked context.

4. What is the difference between x = y++; and x = ++y; ?

Answer: x = y++; uses y's value before incrementing. x = ++y; increments y before using its value.

5. What is the difference between break, continue, and return when used inside a loop statement?

Answer: break exits the loop; continue skips to the next iteration; return exits the method.

6. What are the three parts of a for statement and which of them are required?

Answer: Initializer, condition, iterator; all are optional, but semicolons are required.

7. What is the difference between the = and == operators?

Answer: = assigns a value; == compares for equality.

8. Does the following statement compile? for ( ; true; ) ;

Answer: Yes, it creates an infinite loop.

9. What does the underscore \_ represent in a switch expression?

Answer: Represents the default case.

10. What interface must an object implement to be enumerated over by using the foreach statement?

Answer: IEnumerable or IEnumerable<T>.