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

A person’s telephone number

- String

A person’s height

- float / double

A person’s age

- int

A person’s gender (Male, Female, Prefer Not To Answer)

- enum

```
enum Gender
{
Male,
Female,
Prefer Not to answer
}
```

A person’s salary

- decimal

A book’s ISBN

- long

A book’s price

- decimal

A book’s shipping weight

- float/double

A country’s population

- ulong

The number of stars in the universe

- ulong

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

- int

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

· Value type vs. Reference type:

- Value types store data directly, are allocated on the stack, and copying creates independent copies of the data (e.g., int, float, struct).
- Reference types store references to data, are allocated on the heap, and copying creates new references to the same data (e.g., classes, interfaces).

· Boxing and Unboxing:

- Boxing is converting a value type to object or an interface type.
- Unboxing is extracting a value type from an object or an interface type.

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

· Managed resource: Automatically managed by .NET's garbage collector (e.g., memory allocated by new database connections managed by ADO.NET).

· Unmanaged resource: Not managed by the garbage collector, requiring explicit cleanup (e.g., native handles, file handles not managed by .NET).

4. What’s the purpose of Garbage Collector in .NET?

· Automatically manages memory by reclaiming unused objects.

· Prevents memory leaks by freeing memory no longer in use.

· Optimizes performance by minimizing manual memory management tasks.

Strings

1. When to use String vs. StringBuilder in C#?

- String: Use string when the content is immutable (unchanging), such as storing constant values or text that won't change frequently.
- StringBuilder: Use StringBuilder when you need to dynamically modify or concatenate strings in a loop or when building strings dynamically. It is more efficient for such operations as it minimizes memory allocations and overhead.

2. What is the base class for all arrays in C#?

- The base class for all arrays in C# is System.Array.

3. How do you sort an array in C#?

You can sort an array in C# using the Array.Sort() method, which sorts the elements in an array in ascending order using the default comparer for the element type. Example:

```
int[] numbers = { 3, 1, 4, 1, 5, 9, 2, 6 };
Array.Sort(numbers); // Sorts the 'numbers' array in-place
```

4. What property of an array object can be used to get the total number of elements in an array?

The Length property of an array object can be used to get the total number of elements in the array. Example:

```
int[] numbers = { 1, 2, 3, 4, 5 };
int length = numbers.Length; // length will be 5
```

5. Can you store multiple data types in System.Array?

- Yes, System.Array can store multiple data types. It is a base class for all arrays in C#, and its elements can be of any type derived from System.Object.

6. What’s the difference between System.Array.CopyTo() and System.Array.Clone()?

- Array.CopyTo(Array array, int index): Copies elements from one array to another array. You specify the destination array (array) and the starting index (index) in the destination array where copying begins.
- Array.Clone(): Creates a shallow copy of the array. It returns a new array object with the same length and elements as the original array. Changes to the elements in the clone will not affect the original array, but the elements themselves are still references to the same objects.