

CAT 1

Advanced Application Programming

Wallace Wahong'o 2003944 BSD

- a) Define the following terms as used in C# programming
- i) sealed class
A sealed class is a class that can not be inherited from by other class, prevents an unintended class modification and ensure class's functionality remains as designed.
 - ii) new method
new is keyword used to override method of the same name and signature inherited from the base class.
 - iii) Indexers
Indexers are a special type of property in C# that allows an object to be accessed using square brackets [] notation, similar to an array.
- b) Explain any three limitations of using arrays as opposed to collections.
- Fixed size: arrays have a fixed sized. They can not be modified once created while collections are dynamically re sizable.
 - Limited functionality: Arrays offer basic functionalities like accessing elements by index and iterating through them while Collections provide a richer set of functionalities like searching,
 - Type safety: Arrays are type-safe, meaning they can store only elements of a specified type, can be limiting when you need to store heterogeneous data or work with complex data structures while Collections, on the other hand, can be generic (List<T>) allowing you to store any data type within the specified type parameter T.
- c) Briefly explain the following types of collections
- i). ArrayList ArrayList is a non-generic collection in C# that can dynamically resize and store elements of any data type, including different types of objects. It is part of the System.Collections namespace
 - ii). Hashtable **Hashtable** is a non-generic collection in C# that stores key-value pairs and provides fast retrieval based on the key. It is part of the System.Collections namespace.

- d) Given a list of integers (1, 2, 3, 4), write C# code for inserting these numbers into
i). Array

```
// Define the array
int[] numbers = new int[4];

// Insert numbers using a loop
for (int i = 0; i < numbers.Length; i++)
{
    numbers[i] = i + 1; // Assign values (1, 2, 3, 4)
}

// Print the array
Console.WriteLine("Array: ");
foreach (int num in numbers)
{
    Console.Write(num + " ");
}
```

- ii). ArrayList

```
// Create an ArrayList object
ArrayList numberList = new ArrayList();

// Add numbers using the Add() method
numberList.Add(1);
numberList.Add(2);
numberList.Add(3);
numberList.Add(4);

// Print the ArrayList (optional)
Console.WriteLine("\nArrayList: ");
foreach (int num in numberList)
{
    Console.Write(num + " ");
}
```

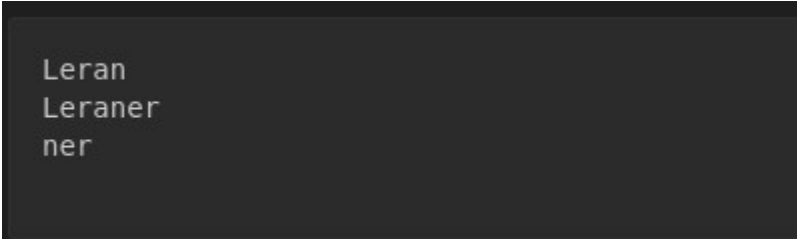
e) Differentiate between `System.Text.StringBuilder` class and `System.String` class

System.String: Immutable, less efficient for repeated modifications, suitable for simple operations, and provides a rich set of methods.

System.Text.StringBuilder: Mutable, more efficient for repeated modifications, suitable for complex string manipulations and performance-critical applications, with a focus on modification methods.

f) What is the output produced by the following program

```
using System;
class DemoString
{
    public static void Main()
    {
        string s1="Lean";
        string s2=s1.Insert(3,"r");
        string s3=s2.Insert(5,"er");
        string s4=s3.Substring(4);
        Console.WriteLine(s2);
        Console.WriteLine(s3);
        Console.WriteLine(s4);
    }
}
```

A screenshot of a console window showing the output of the C# program. The output consists of three lines: "Leran", "Leraner", and "ner".

```
Leran
Leraner
ner
```