Data Structure Prelim Lab Exam

Java| BSCS 2A

September 09, 2025| Lab 06 5:30PM - 08:30PM

Instructions:

- Write clean, properly indented Java code.
- Complete the given code based on the comments
- Create a folder named ***PrelimLab***. This is where you should place your java files.
- Observe proper Method and Variable casing.
- *Proper & Correct FileName is part of the exam.*
- *Public class dictates the filename*

Question 1: Array. (10 points)

```
public class SimpleArray {
    public static void main(String[] args) {
        int[] numbers = new int[5];
        // TODO: Store the numbers 10, 20, 30, 40, 50 in the array
        // TODO: Print all values using a for loop

    }
}
```

Question 2: Array List (20 points)

```
public class MyList {
    // TODO: Declare a private array of String named items
    // TODO: Declare a private int variable named currentIndex
    // TODO: Declare a private int variable named maxCapacity

    // TODO: Create a constructor that accepts size
    // Inside constructor:
    //    - initialize items with new String[size]
    //    - initialize maxCapacity to size
    //    - initialize currentIndex to 0

    public void add(String value) {
        // TODO: Add the given value at the end of the list
        //  IF full print "List is full"
    }
    public void remove(int index) {
        // TODO: Remove element at the given index
        // Shift all elements to the left
        // No Empty or null should appear during printing
        // Decrease size
        // Below code is almost correct, What should be place in the <code here>
        for(int a = index; a < size - 1; a++) {
            items[a] = <code here>;
        }
                // We are actually not removing, we just replaced the deleted item by the  next item
    }
    public void modify(int index, String newValue) {
        // TODO: Change the element at the given index to newValue
    }
```

```java
    public void display() {
        // TODO: Print all elements in vertical format
        // Do not print empty or null elements, | Must not exceed the current index
        // Do not skip index should be in order
        // Example output:
        // Item #1: Juan
        // Item #2: Maria
        // Item #3: Pedro
    }

    public static void main(String[] args) {
        // TODO: Create an object of MyList with capacity 5
        // TODO: Add 3 names into the list
        // Example: "Juan", "Maria", "Pedro"
        // TODO: Display all names
        // TODO: Add 1 more name and display again
        // TODO: Modify the 2nd name to "Carlos" and display again
        // TODO: Remove the 1st item and display again
    }
}
```

Question 3: Stack (20 points)

```
public class MyStack {
    // TODO: Declare a private string array named stack
    // TODO: Declare a private int variable named top

    // TODO: Create a constructor that accepts capacity
    // Inside constructor:
    //    - initialize stack with new int[capacity]
    //    - set top = -1

    public void push(String value) {
        // TODO: Insert value on top of the stack
        // Increase top
        // print Stack is full if top exceeds capacity
    }

    public int pop() {
        // TODO: Remove and return the top value
        // Decrease top
        // Print Stack is empty if top is -1
    }

    public int peek() {
        // TODO: Return the top value without removing it
        // Print Stack is empty if top is -1
    }

    public void display() {
        // TODO: Print all elements from top to bottom
        // Example:
        // Top -> 50 40 30 <- Bottom
    }

    public static void main(String[] args) {
        // TODO: Create a stack with capacity 5
        // TODO: Push 3 numbers (10, 20, 30)
        // TODO: Display the stack
        // TODO: Peek the top number
        // TODO: Pop 1 number and display again
    }
}
```

Question 4: Linked List (25 points). The same file

```
class Node {
    // private  declare char data
    // public declare Node next

    // One constructor here that accepts a char parameter

    // public getData that returns char data
    // public setData that sets char data
}


public class MyLinkedList {
    // TODO: Declare a private Node variable named head
    // TODO: Declare a private int variable named size

    public MyLinkedList() {
        // TODO: Initialize head = null and size = 0
    }

    public void add(String value) {
        // TODO: Create a new node
        // If head == null, assign it as head
        // Else, traverse to the end and link the new node
        // Increase size
    }

    public void display() {
        // TODO: Traverse from head to null
        // Print each node's data in vertical format
        // Example:
        // Node #1: Juan
        // Node #2: Maria
    }

    public void remove(String value) {
        // TODO: Remove the first node that matches the value
        // Adjust links so the list remains connected
        // Decrease size if removed
    }

    public void modify(String oldValue, String newValue) {
        // TODO: Traverse the list to find the node
        // If a node matches oldValue, change its data to newValue
         //code here
         Node current = this.head;
         while(current != ?){
            if(){
                // replace the value
                return;
            }
            current = current.?
         }

    }

    public static void main(String[] args) {
        // TODO: Create an object of MyLinkedList

        // TODO: Add 3 names into the list
        // Example: "Juan", "Maria", "Pedro"
```

```
        // TODO: Display all nodes

        // TODO: Add 1 more name and display again

        // TODO: Modify "Maria" to "Carlos" and display again

        // TODO: Remove "Juan" and display again
    }
}
```