Lab 11
Singly List Link

Submission Rules:

1. Submissions must be zipped into a **handin.zip** file. Each problem must be implemented in its own class file. Use the name of the problem as the class name.

2. You must use standard input and standard output for ALL your problems. It means that if input is needed the input should be entered from the keyboard while the output will be displayed on the screen.

3. Your source code files should include a comment at the beginning including your name and that problem number/name.

4. The output of your solutions must be formatted exactly as the sample output to receive full credit for that submission.

5. Compile & test your solutions before submitting.

6. Each problem is worth 40 points total.

7. This lab is worth a max total of: 40 points.

8. Submission:

- You have unlimited submission attempts until the deadline passes
- You'll receive your lab grade immediately after submitting

Problem 1: Singly Linked List (40 points)

(Data Structures) A singly linked list creates a data structure in which each node only knows which node is the next node in the data structure. A node only contains data and a reference to the next node.

UML Class Diagram:

SinglyLinkedList
- listName: String - head: Node
• add(): void • getByIndex(): int • addByIndex(): void • removeByElement(): void • removeByIndex(): void

DateFormatConverter **Method API:**

Modifier and Type	Method and Description
public void	add(int e) Inserts the specified integer primitive element e into the end of the SinglyLinkedList.
public int	getByIndex(int index) Returns the value contained inside the specified index. If the given index is less than 0 then the method will return -999.
public void	addByIndex(int index, int element) Inserts the specified element into the SinglyLinkedList at the specified index.
public void	removeByValue(int value) Removes the element of the specified value from the SinglyLinkedList.
public void	removeByIndex(int index) Removes the element at the specified index from the SinglyLinkedList.