**JOBSHEET 11**

**Linked List**



**Name**

Sherly Lutfi Azkiah Sulistyawati

**NIM**

2341720241

**Class**

1I

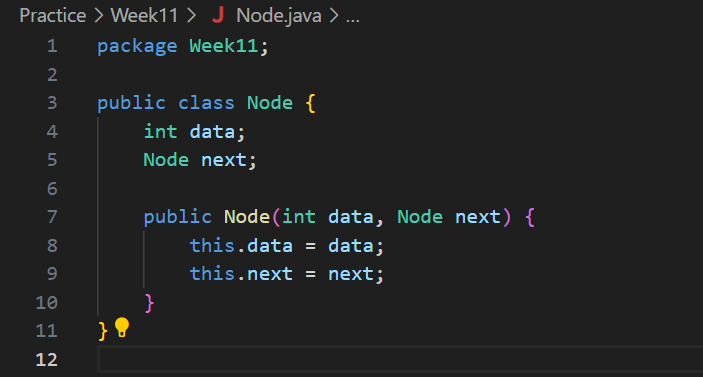
**Major**

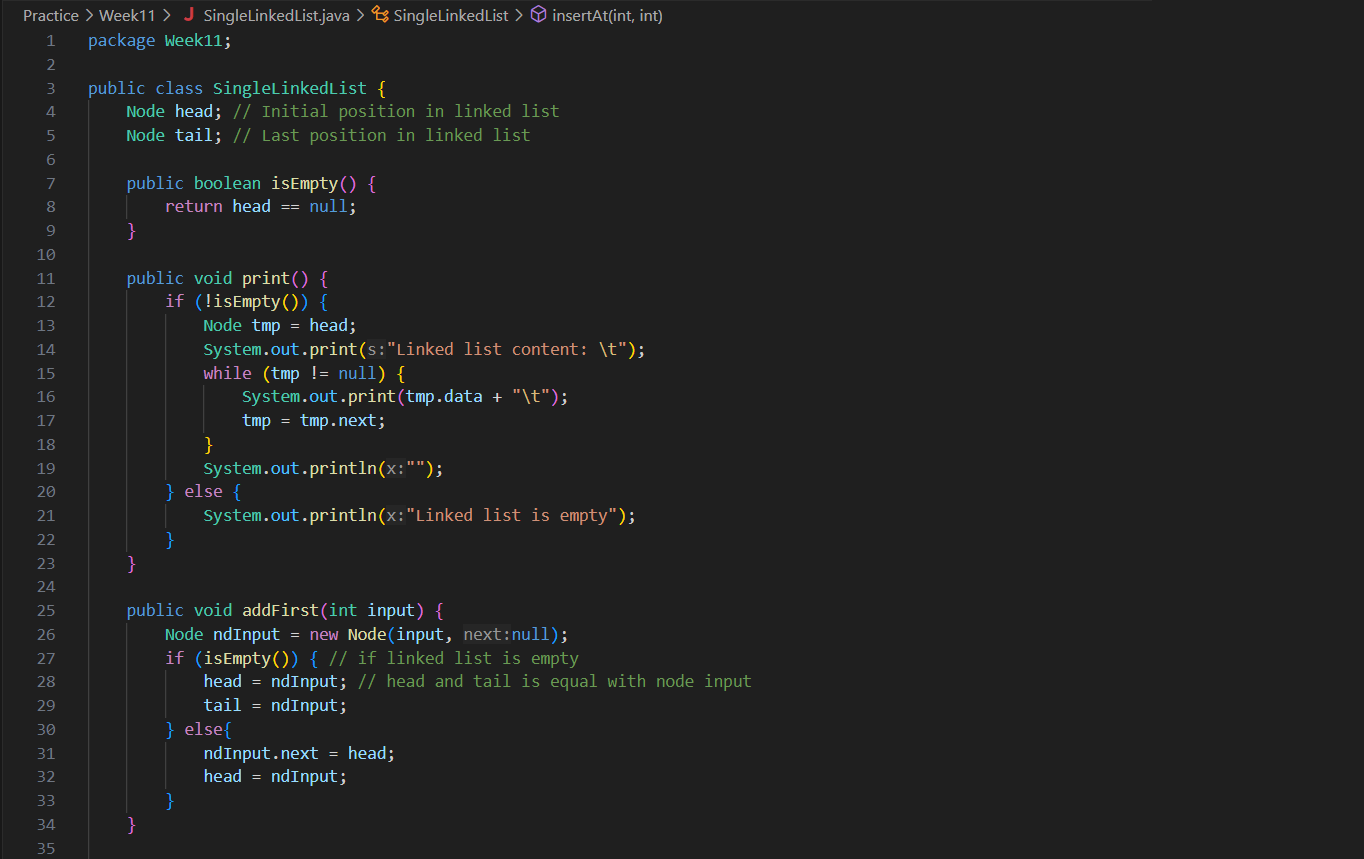
Information Technology

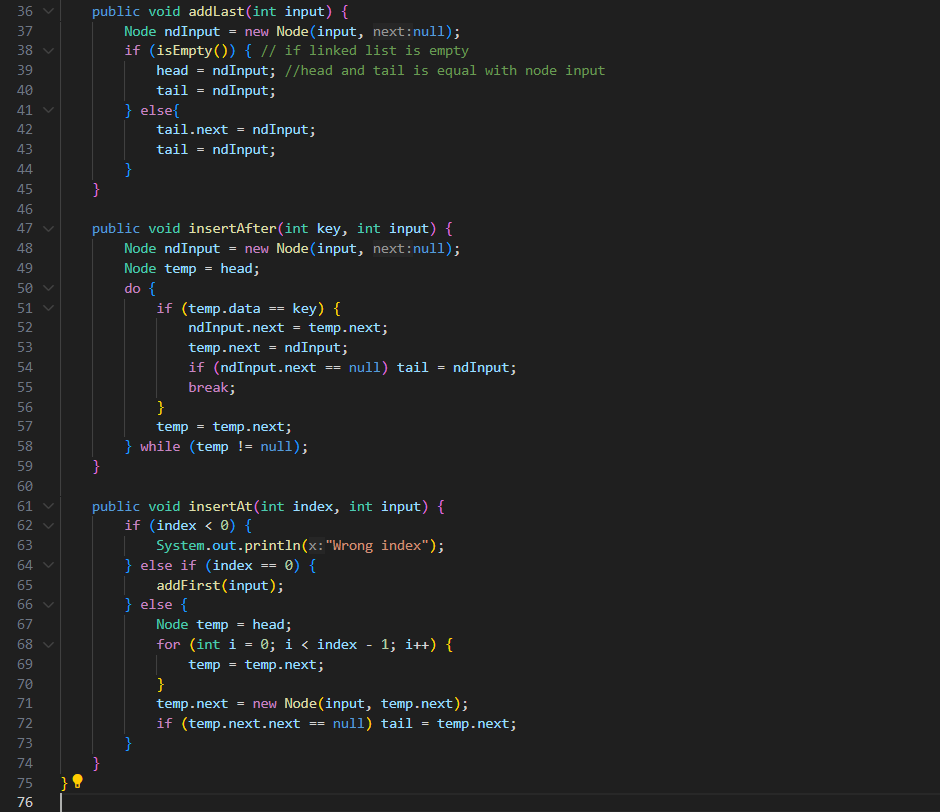
**Study Program**

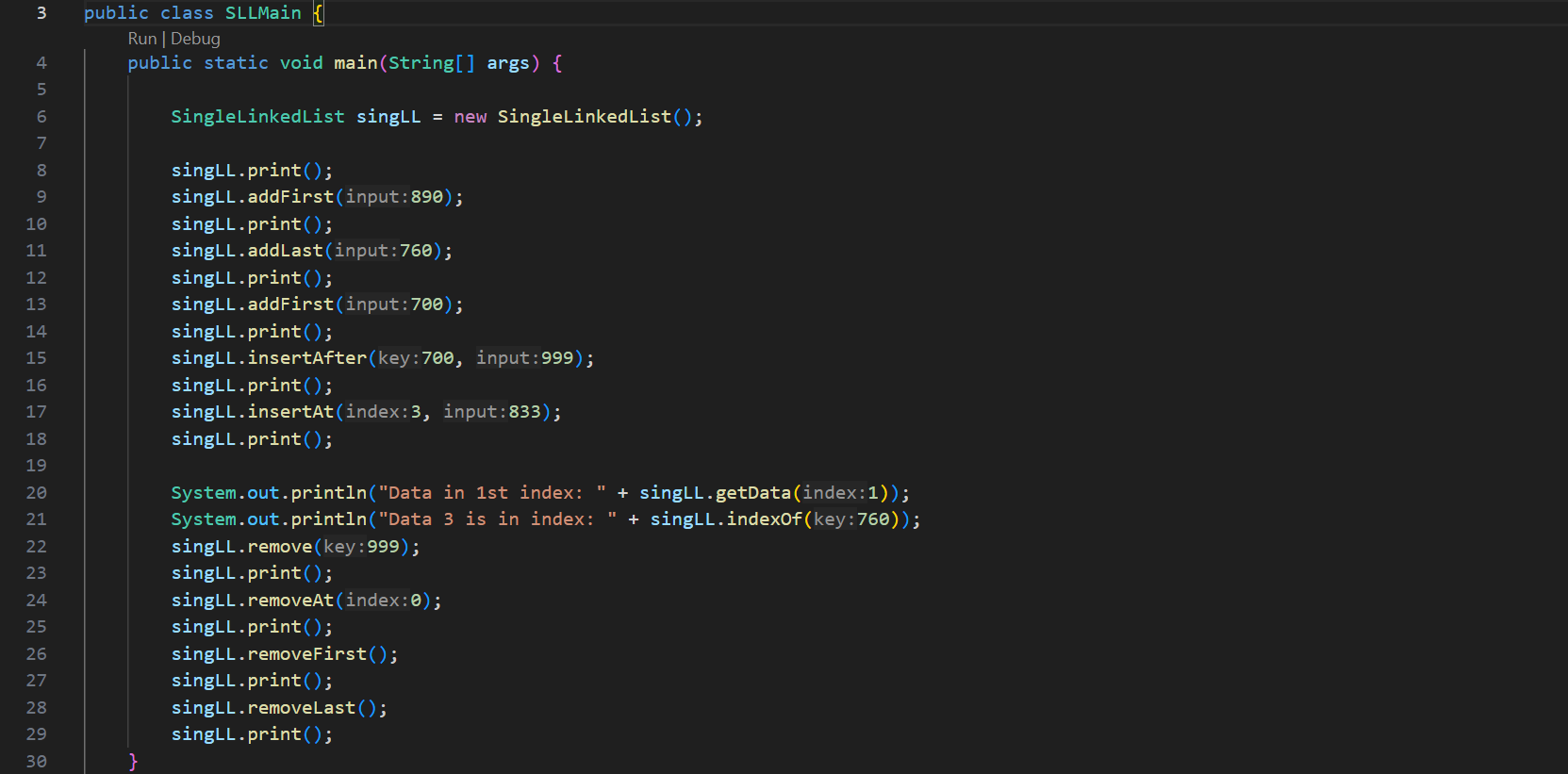
D4 Informatics Engineering

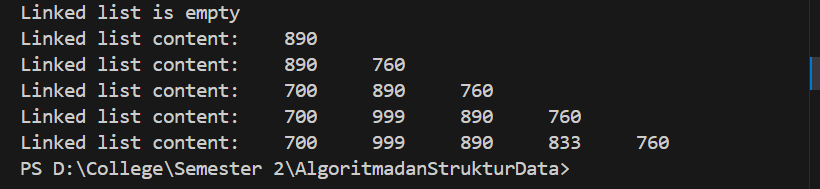
**Lab Activity 1**

****

****

****

****

****

**Question**

1. Why the output of the program in first line is “Linked list is empty”?

- The output is "Linked list is empty" because the linked list is empty when the print() method is first called. No elements have been added to the list at that point.

1. Please explain the usage of these following codes in:



- These lines of code are used in the insertAfter method to insert a new node (ndInput) after a specified node (temp) with a certain key value in the linked list.

- ndInput.next = temp.next;: Links the new node (ndInput) to the node that follows temp.

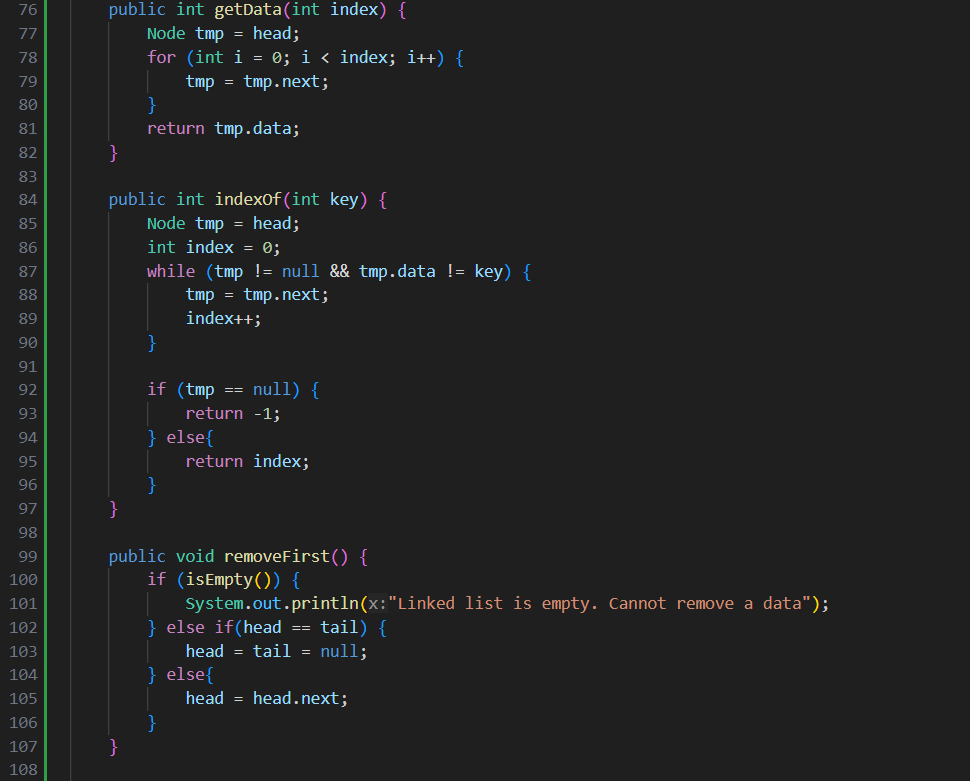
- temp.next = ndInput;: Links the temp node to the new node (ndInput), inserting ndInput after temp.

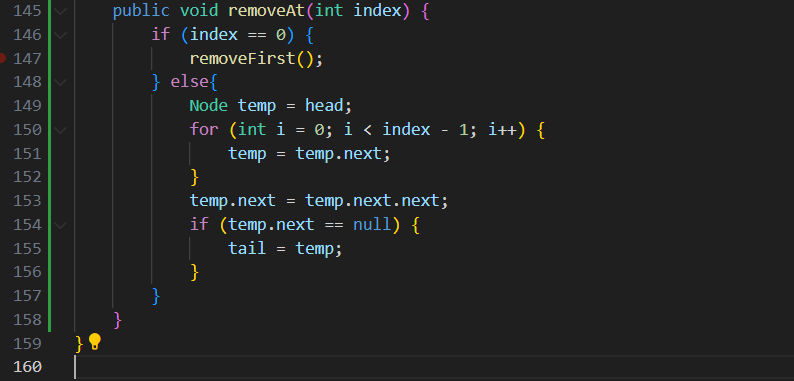
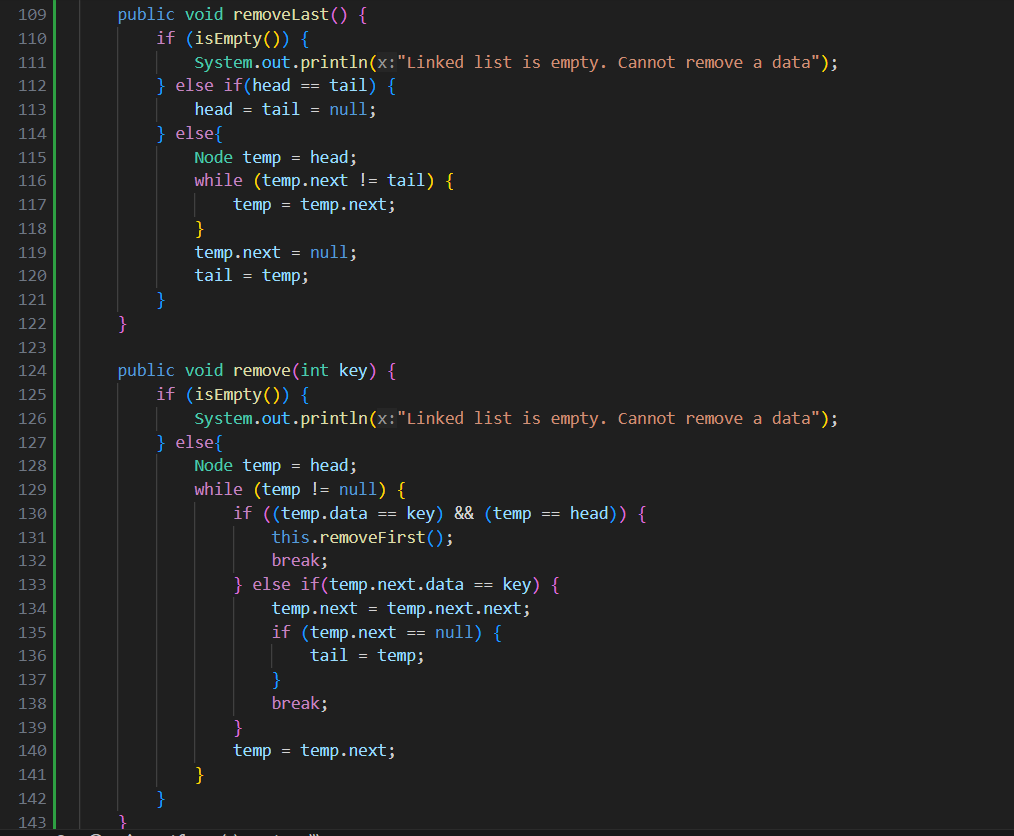
1. In SingleLinkedList, what is the usage of this following code in insertAt?

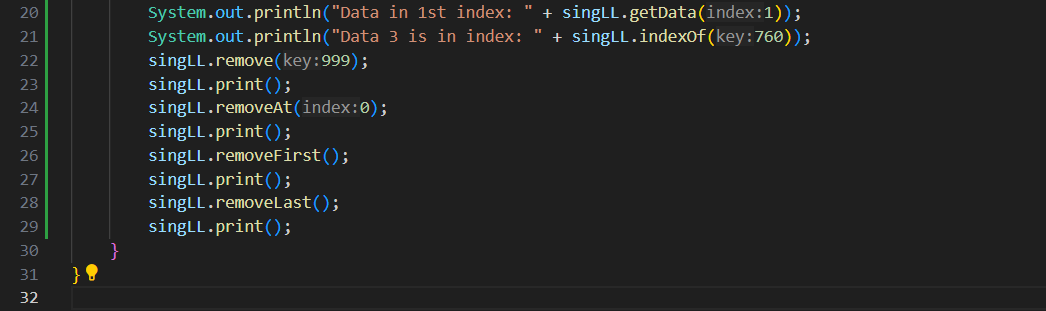


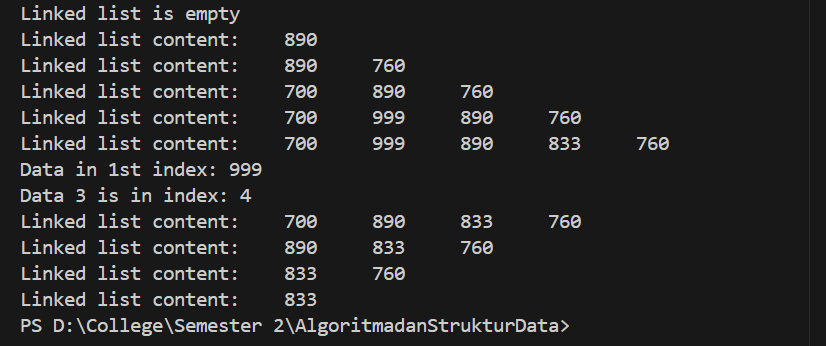
- This line of code is used to update the tail pointer if the new node is inserted at the last position of the linked list. (i.e., the new node's next is null).

**Lab Activity 2**

****

****

****

****

**Question**

1. Why we use break keyword in remove function? Please explain

- The break keyword is used to exit the while loop immediately after the node with the specified key is found and removed, preventing further unnecessary iterations.

1. Please explain why we implement these following codes in method remove



- This code handles the case where the node to be removed is not the head but is somewhere else in the list.

- temp.next.data == key: Checks if the data of the node immediately following temp is equal to the key.

- temp.next = temp.next.next;: Bypasses the node to be removed by setting the next pointer of temp to point to the node after the node to be removed. This effectively removes the node with the key from the list.

1. What are the outputs of method indexOf? Please explain each of the output!

- The indexOf method returns the index of the first occurrence of a node with the specified key. If the key is not found, it returns -1.

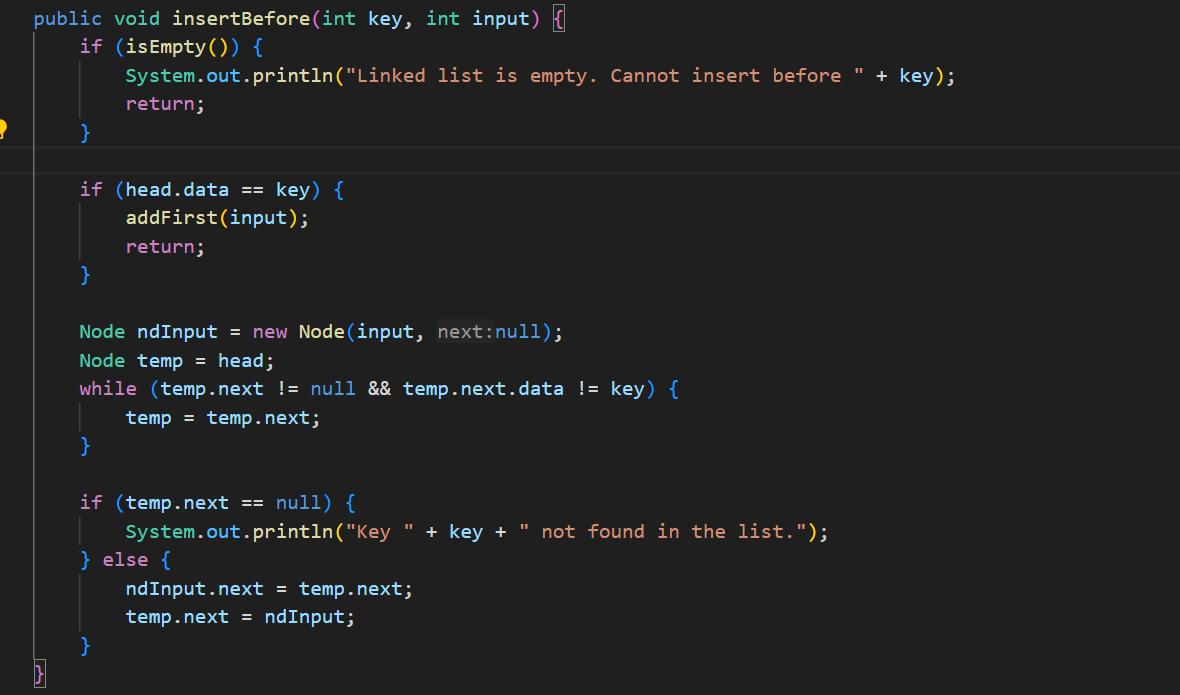
- Example outputs:  
If the list is [700, 890, 760] and we call indexOf(890), it will return 1 because 890 is at index 1.

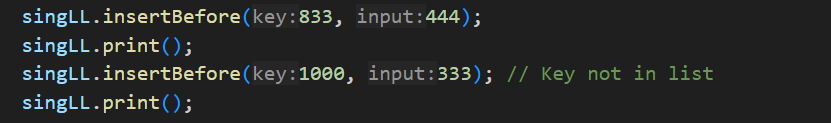
If the list is [700, 890, 760] and we call indexOf(760), it will return 2 because 760 is at index 2.

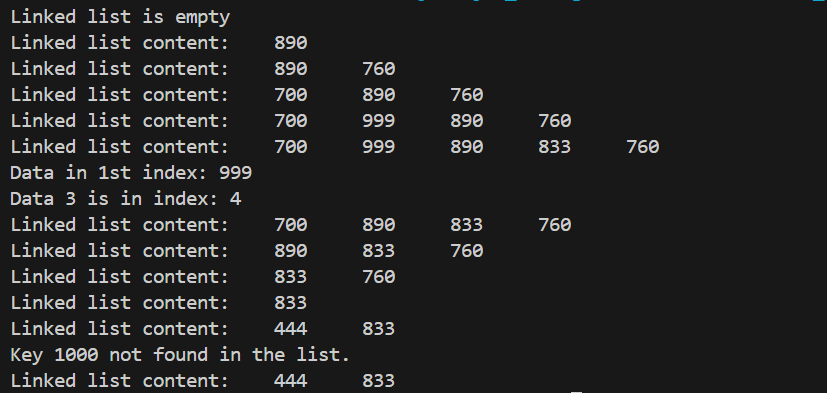
If the list is [700, 890, 760] and we call indexOf(999), it will return -1 because 999 is not in the list.

**Assignment**

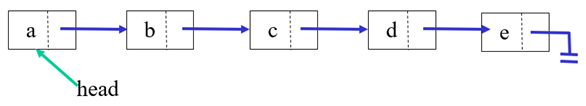
1. Create a method **insertBefore()** to add node before the desired keyword

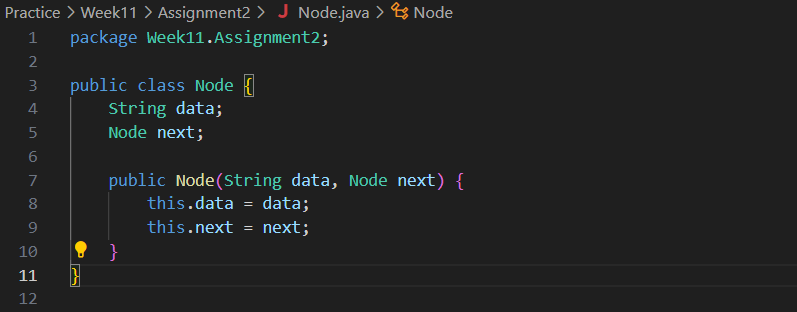


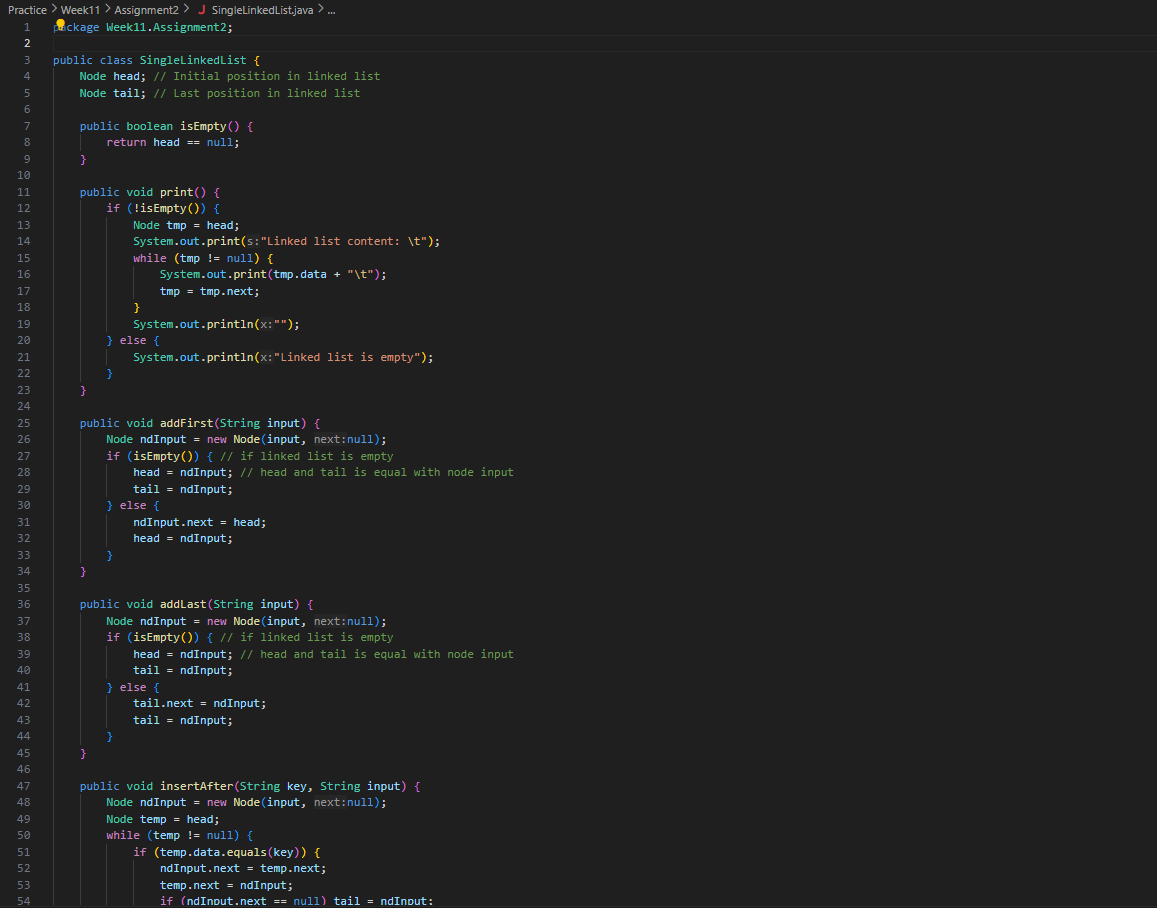


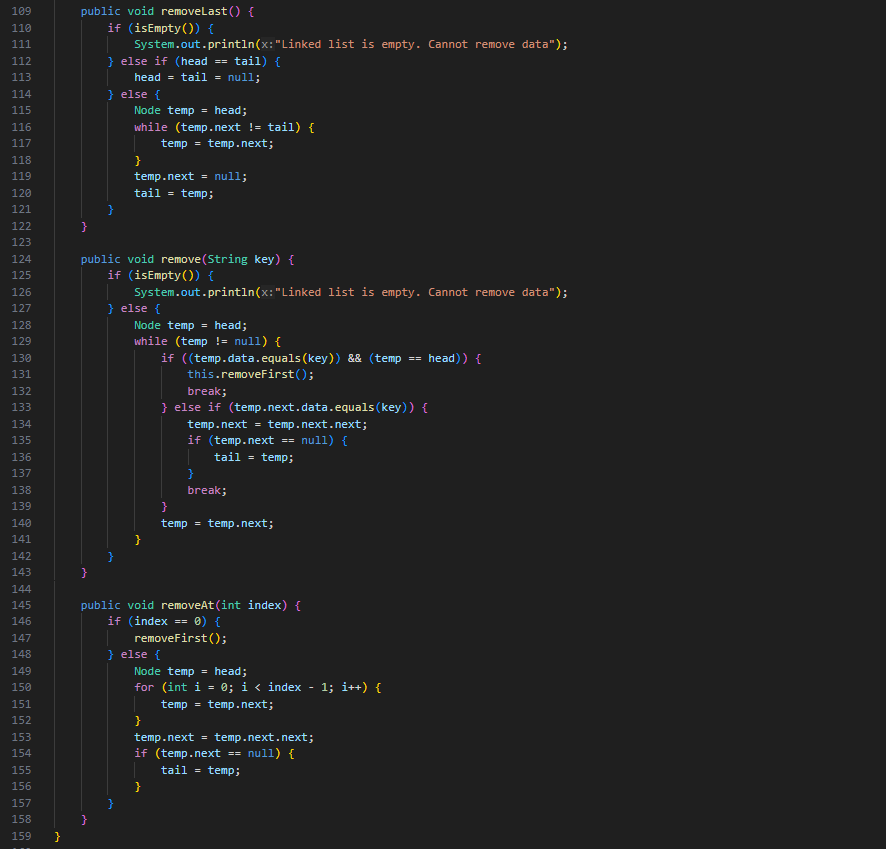
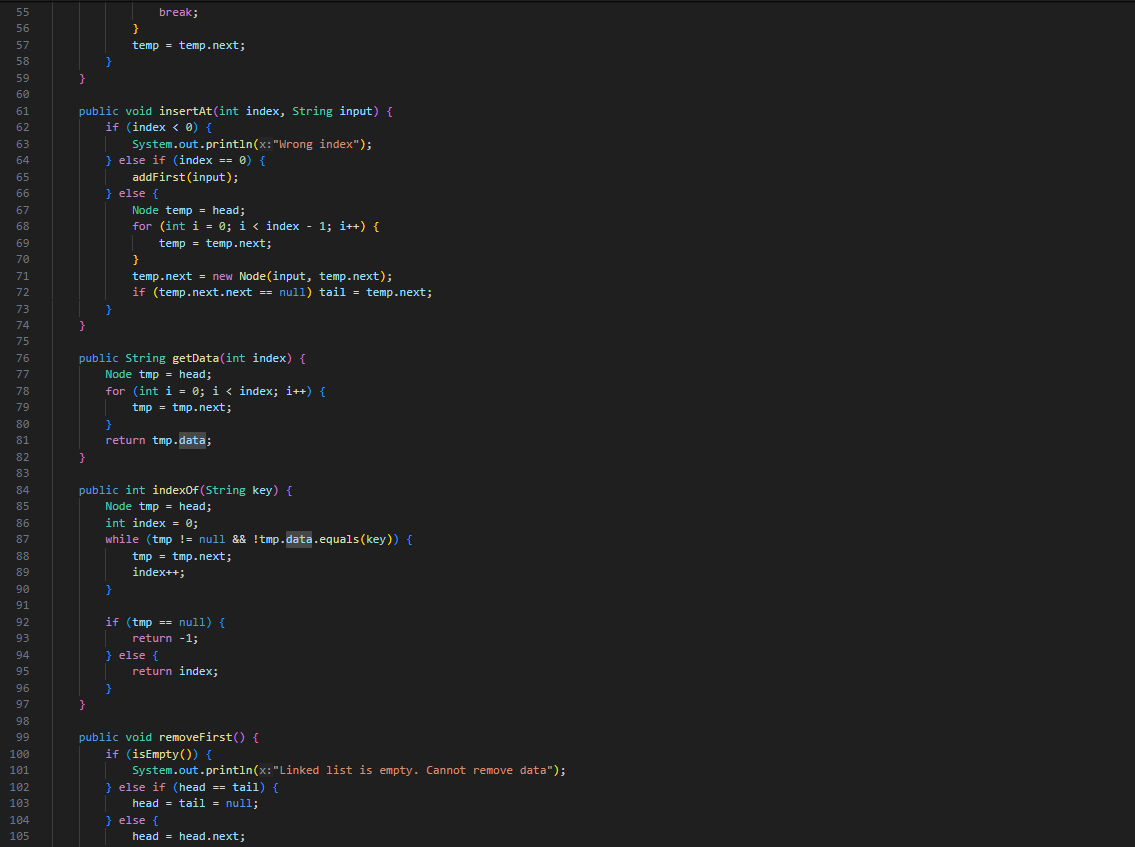


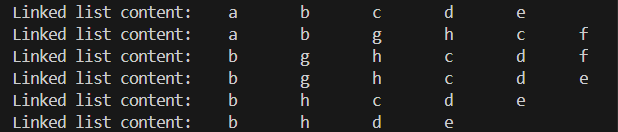
1. Implement the linked list from this following image. You may use 4 method of adding data we’ve learnt



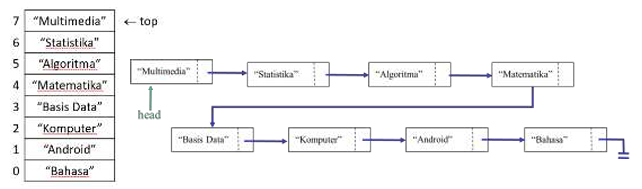


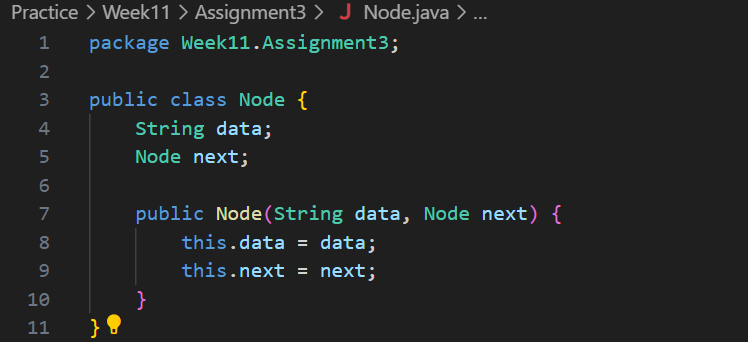


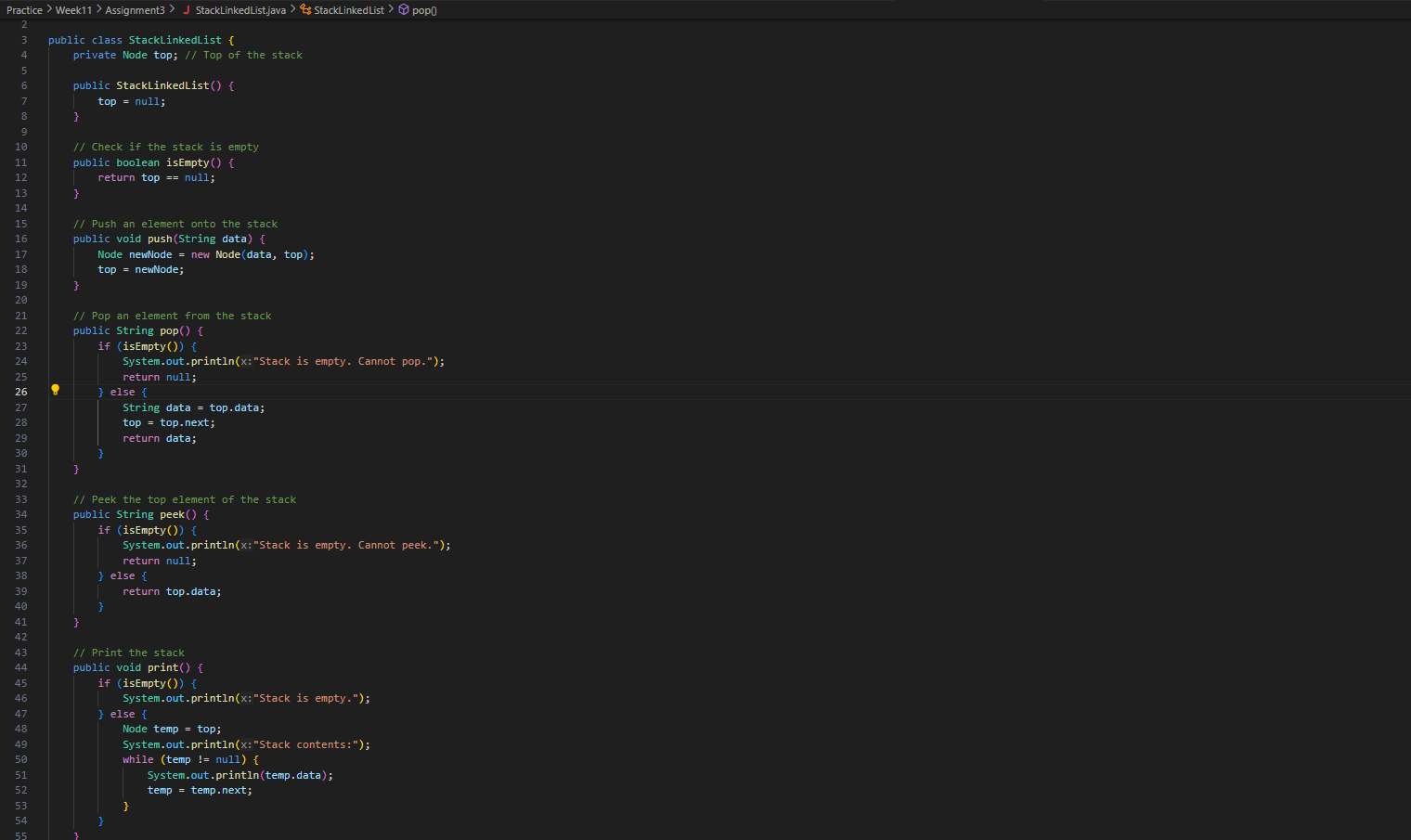


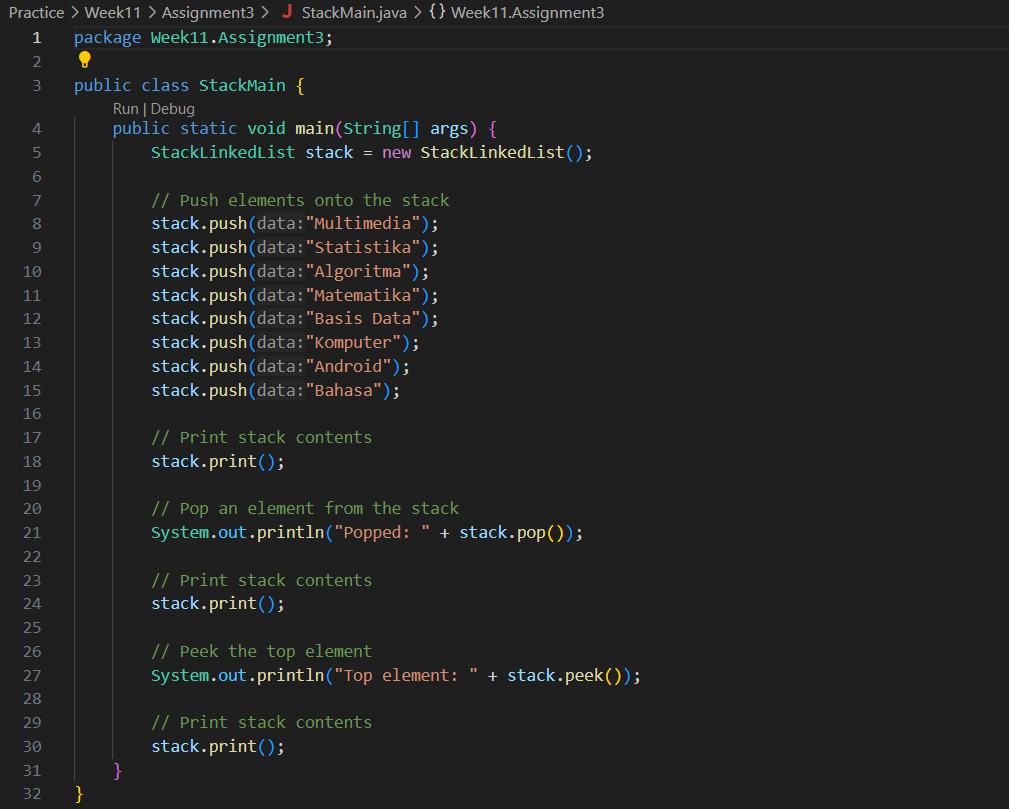


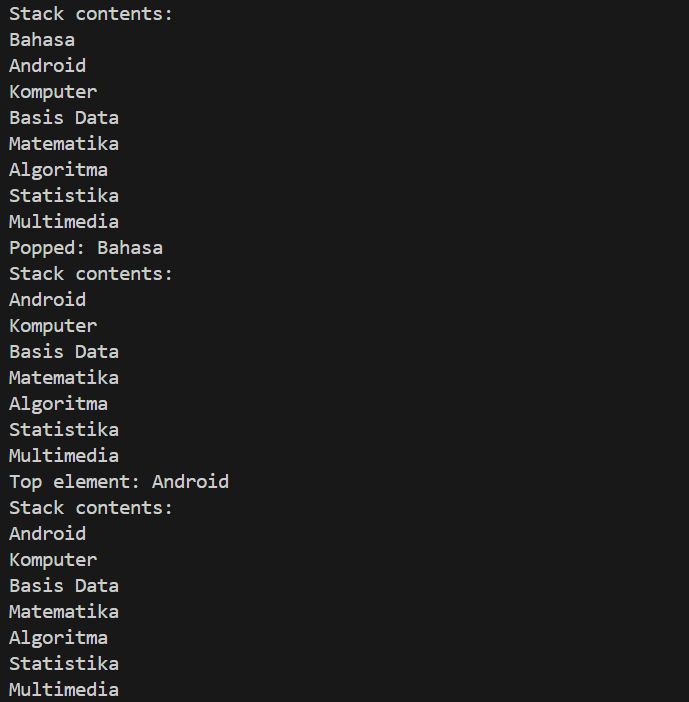
1. Create this following **Stack** implementation using Linked List implementation



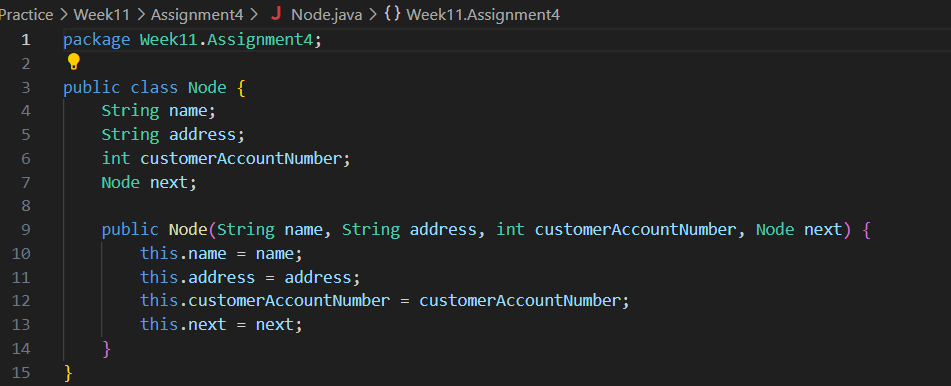


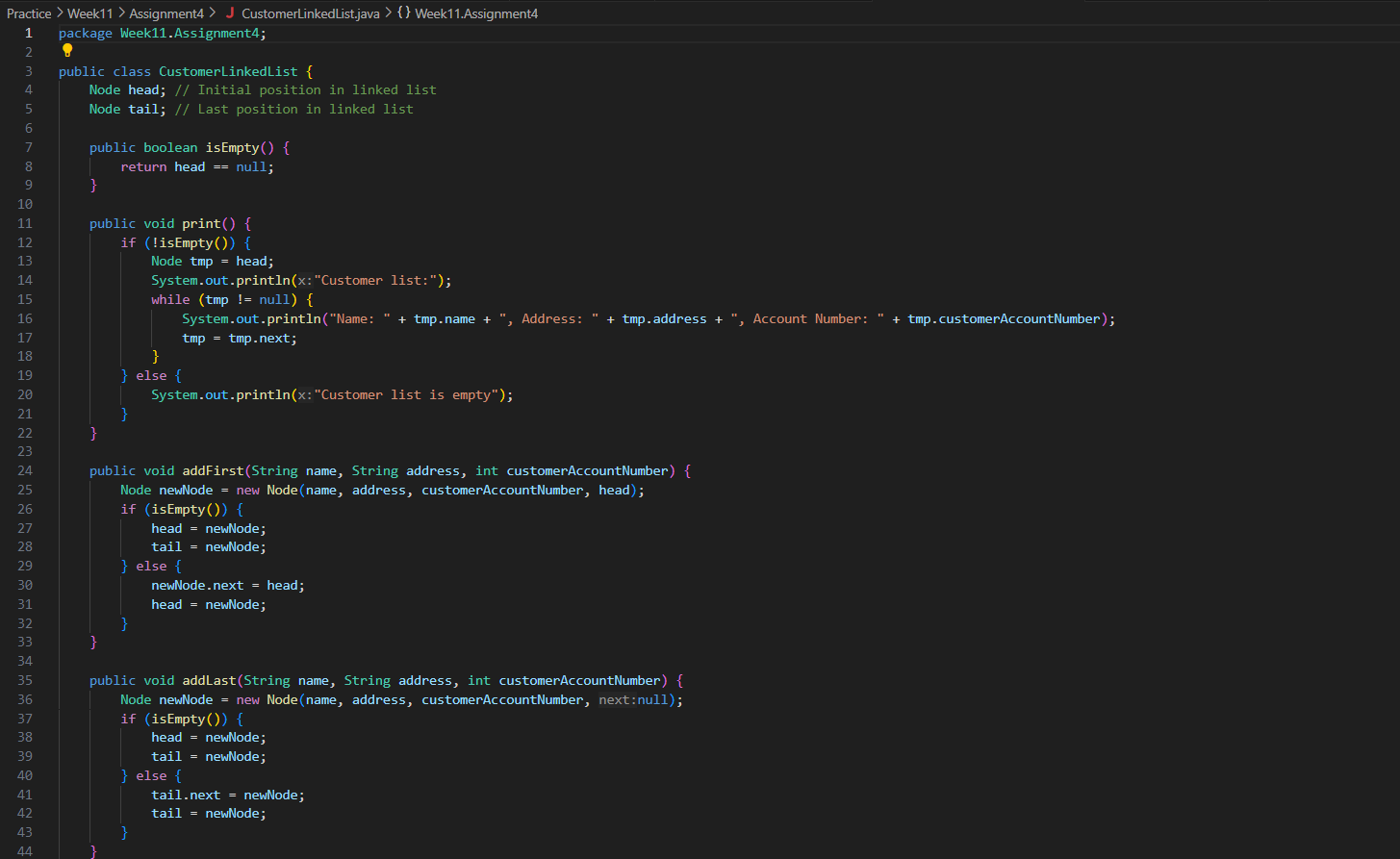
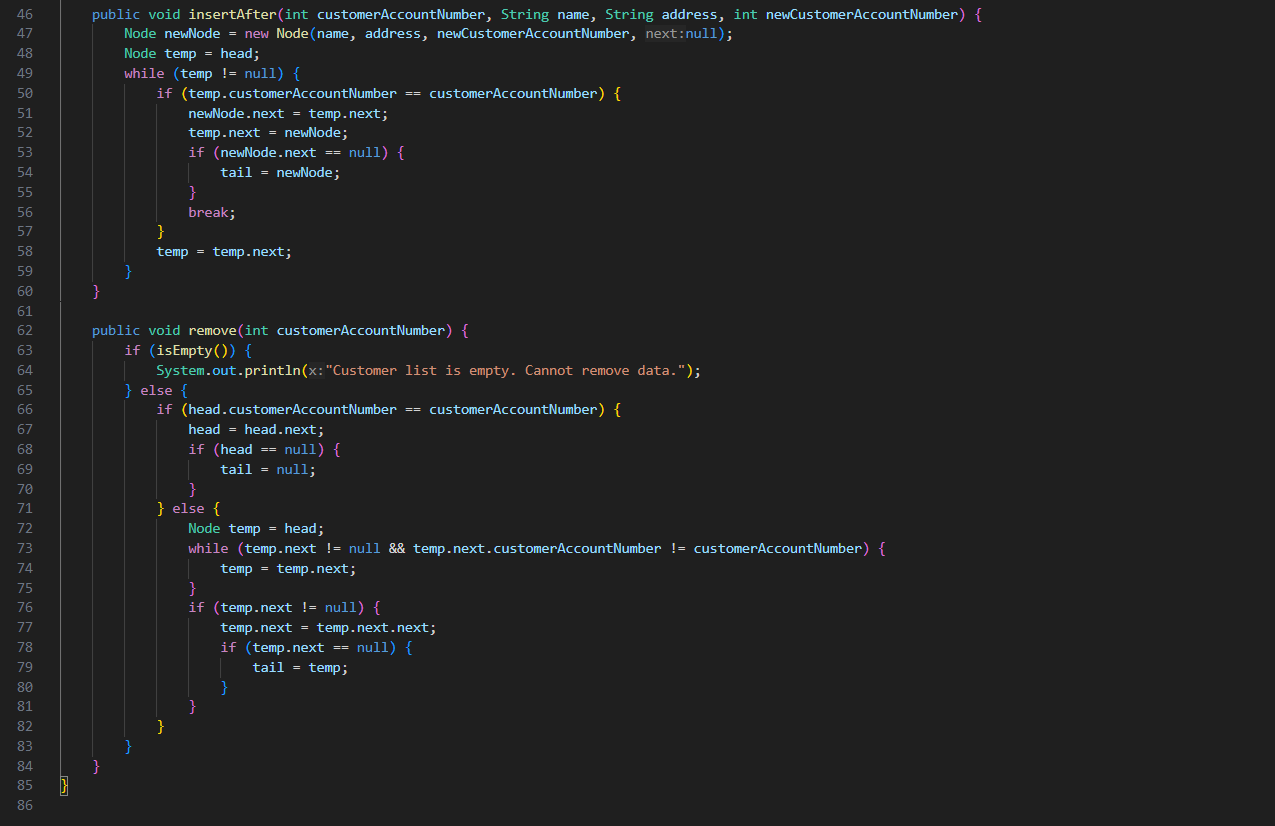




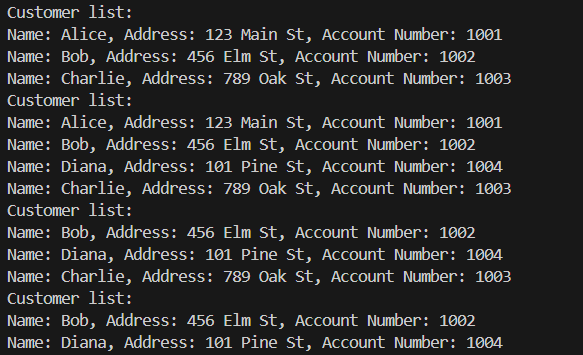


1. Create a program that helps bank customer using linked list with data are as follows: Name,address, and customerAccountNumber







1. Implement **Queue** in previous number with **linked list** concept

