The second quiz for the course will be on Monday, Feb 19, during your lab session. The following list provides more information about the quiz:

- The quiz will be a written quiz (no computer).
- Closed book, closed notes quiz.
- Answers must be neat and legible. We recommend that you use pencil and eraser.

The following exercises cover the material to be included in this quiz. Solutions to these exercises will not be provided, but you are welcome to discuss your solutions with the TA or instructor during office hours. **We strongly recommend you do not use Eclipse to write the code associated with these exercises.** Try to answer the exercises in a piece of paper and then use Eclipse to verify your solutions. This approach will better prepare you for the quiz. **You cannot use any Java API class (except String) during the implementation of the methods below.**

**Exercises**

Implement the methods below based on the following Java class definitions.

```java
public class Node {
    String data;
    Node next;
}

public class LinkedList {
    Node head;
}
```

1. Define a constructor for the LinkedList class that creates an empty list.
2. Define a method called `addFirst` that adds a String to the beginning of the list.
3. Define a method called `addLast` that adds a String to the end of the list.
4. Define a method named `size` that returns the number of nodes in the list.
5. Implement a method named `print` that prints the String data of each node in the list.
6. Implement a method named `removedLastNode` that removes the last node from the list.
7. Define a method named `find` that has as parameter a String reference. The method will return true if the String is found in the list and false otherwise. Use the `equals` method to compare Strings.
8. Define a method called `insert` that has the following prototype:
   ```java
   public boolean insert(String entry, String target);
   ```
   The method will look for the first instance of `target` in the list and will insert `entry` before that instance. The method will return true if a `target` instance is found and false otherwise. The list should not be modified if a `target` instance is not found.
9. Define a method called `filter` that has the following prototype:
   ```java
   public LinkedList filter(int maxLength);
   ```
   The method will return a new LinkedList with nodes containing only strings that are shorter or equal to `maxLength`. The method should not modify the original list.