CMPSCI 187 (Spring 2018) Lab 02: Linked Lists

The goal of this lab is to get some practice using linked lists. To work on the assignment:

- Go to ***File -> Make a Copy*** to make an editable copy of this Google Doc for your account
- Follow the instructions to complete the assignment
- When you are done, go to ***File -> Download As -> PDF Document***
- Log in to Gradescope and submit your PDF

The code you are given is:

- A Dog class, where a Dog object contains a name and a weight,
- An LLDogNode class, whose objects are linked list nodes each containing a Dog, and
- A stub DogTeam class, in which you will complete four methods.

Exercise:

A DogTeam contains one or more Dog objects in some order. Your task is to implement the following four methods:

1. insertHead, which puts a new Dog at the head of the list,

2. insertTail, which puts a new Dog at the tail of the list, and

3. weightDiff, which returns the difference between the weights of the heaviest and the lightest Dog in the list. (A sled dog team is more effective if all the dogs in it have close to the same weight.)

4. insertAfter, which puts a new Dog in the list after the Dog with name dogName. (The exercise assumes that there is only one Dog object with a given name, so you do not have to worry about multiple dogs having the same name.)

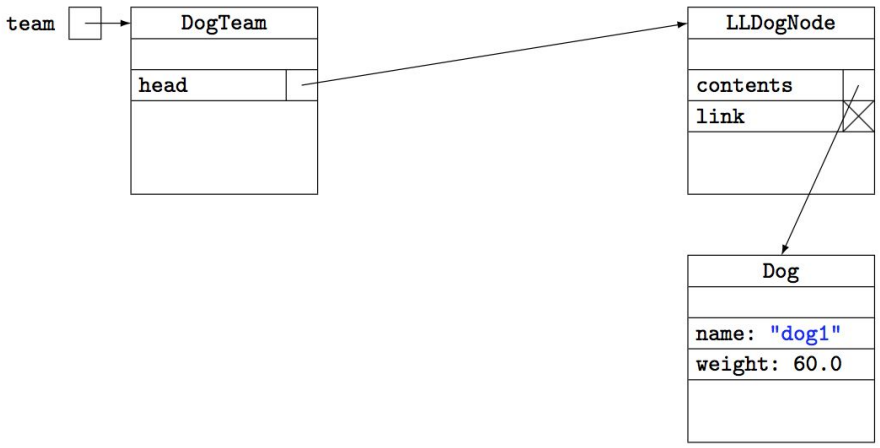
To help you understand how the linked lists play a role in this exercise we provide a pictorial representation of the main method.

```
22
23  public static void main(String[] args) {
24
25    DogTeam team = new DogTeam(new Dog("dog1", 60));
26    team.printTeam();
27    System.out.println("weightDiff: " + team.weightDiff());
28
29    team.insertTail(new Dog("dog0",  5));
30    team.insertHead(new Dog("dog2",  90));
31    team.printTeam();
32    System.out.println("weightDiff: " + team.weightDiff());
33
34    team.insertHead(new Dog("dog3",  7));
35    team.insertTail(new Dog("dog4",  100));
36    team.insertTail(new Dog("dog10", 205));
37    team.printTeam();
38    System.out.println("weightDiff: " + team.weightDiff());
39
40  }
```

(End of page 1)

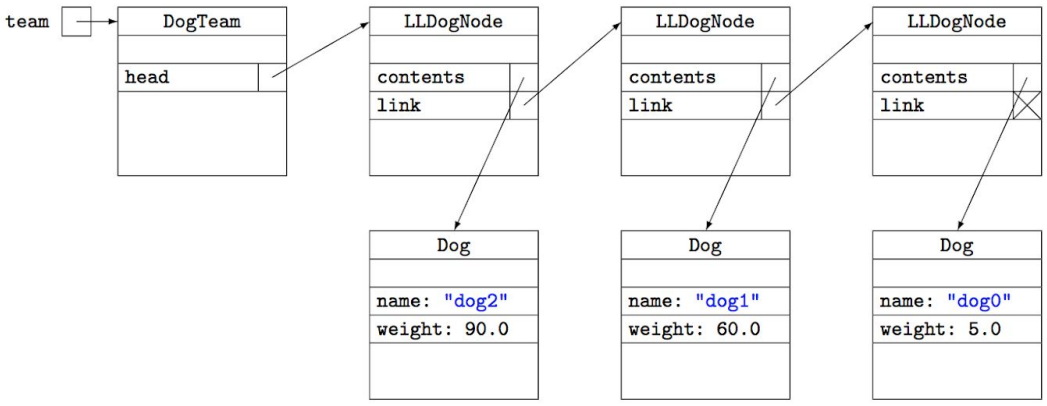
After line 25, we create a DogTeam linked list, where the head of the list points to a dog node containing “dog 1”.

After line 25 of `DogTeam`.



After line 30, we have inserted a node containing “dog 0” at the tail and a node “dog 2” at the head. After these procedures are executed, the list looks as follows:

After line 30 of `DogTeam`.



(End of page 2)

Complete the code to provide the functionality of the linked list. When traversing the list, be aware of possible infinite loops or null pointer exceptions. If you run into any of this, USE THE DEBUGGER! **:**

TODO Method 1

```
public void insertHead(Dog dog){
        LLDogNode node = new LLDogNode(dog, head);
        this.head = node;
}
```

TODO Method 2

```
public void insertTail(Dog dog){
        LLDogNode current = head;
        while (current.getLink() != null) {
                current = current.getLink();
        }
        current.setLink(new LLDogNode(dog, null));
}
```

(End of page 3)

TODO Method 3

```
public void weightDiff(){
	double max = head.getContents().getWeight(), min = max, weight = 0.0;

	LLDogNode current = head;
	while (current.getLink() != null) {
		current = current.getLink();
		weight = current.getContents().getWeight();
		if (weight > max) {
			max = weight;
		}
		if (weight < min) {
			min = weight;
		}
	}

	return max - min;
}
```

TODO Method 4

```
public void insertAfterI(Dog dog, String dogName){
	LLDogNode current = head;
	String name = current.getContents().getName();

	while (current.getLink() != null && !name.equalsIgnoreCase(dogName)) {
		current = current.getLink();
		name = current.getContents().getName();
	}
	current.setLink(new LLDogNode(dog, current.getLink()));
}
```

(End of page 4)

TODO Survey

Complete a short survey to give feedback about the lab at:
https://goo.gl/forms/ChIVATjREIsBuPxl2