

This handout is for this week's recitation, recitation 08, on interfaces Iterator and Iterable. Complete this exercise and show it to your TA by Thursday midnight of this week. Your TA will record that you did it.

IT IS BEST TO COME TO YOUR NORMAL RECITATION THIS WEEK AND DO IT IN THE RECITATION. In the recitation, you can get the help of your TA and others sitting around you. You should be able to finish in the 50 minutes, if you watched the 13.5 minutes of video and did the little exercise in it. And perhaps if you prepare Eclipse beforehand as discussed on the next page.

BRING A LAPTOP TO THE RECITATION IF YOU CAN. IT IS NECESSARY. In the recitation, you can do this work with another person who is sitting next to you. If you don't do it in recitation, do it alone.

The main task in this recitation is for you to take a version of A3, doubly linked lists, and have it implement interfaces Iterator and Iterable. But first a few problems for you to do.

1. Suppose you had to explain to a friend the difference between interfaces Iterator and Iterable. What would you say?

A. Write down in a sentence or two what interface Iterator is for.

B. Write down in a sentence or two what interface Iterable is for.

2. The code below is used to enumerate and print the Strings in some collection. To the right, write down what is wrong with this code:

```
Iterator<String> it= ...;
...
while (it.next() != null) {
    System.out.println(it.next())
}
```

3. The code below is used to enumerate and print the Strings in some collection. To the right, write down a requirement on the collection for this to actually work and also state when it doesn't work.

```
Iterator<String> it= ...;
...
while (it.hasNext() != null) {
    System.out.println(it.next());
    System.out.println(it.next());
}
```

4. This problem concerns implementing Iterator and Iterable in your solution to A3, class DLinkedList. Do the following steps to prepare for this—you can use either your solution to A3, if you know it is correct, or our solution, which is on the course website, page lectureNotes, on the row for recitation 08. Here's the URL:

www.cs.cornell.edu/courses/CS2110/2016sp/recitations/recitation08/a3Solution.zip

- A. Start a new Eclipse project, call it something like a3Iterable.
- B. Fix the project:
 - a. To use YOUR DLinkedList.java: Copy your DLinkedList.java from your old a3 project to the new one. You can do this by selecting the old file, doing a copy, selecting the new project's src directory, and doing a paste. Download the above-mentioned file a3Solution.zip, unzip it, and drag DLinkedListTest.java into the new project, putting it in the default directory.
 - b. To use OUR classes: Get our files (see above) and drag them into the project's src directory.
- C. If JUnit4 is not available in the new project, insert a new JUnit test-case class in the usual way and then delete it.
- D. Look at the code at the bottom of class DLinkedListTest. Look at how it tests interfaces Iterator and Iterable. Run the testing class. There should be no errors. This works because Iterator and Iterable are implemented in DLinkedList's superclass.
- E. Delete the clause "**extends** java.util.AbstractList<E>" on the declaration of DLinkedList<E>. Then delete all the `@Override` annotations. After that, there should be no syntax errors in DLinkedList. But DLinkedListTest will show errors because Iterator and Iterable are now no longer available.
- F. Write inner class DLLIterator with this specification and header:

```
/** An instance is an iterator over the elements of this list. */  
private class DLLIterator implements Iterator
```
- G. Write method iterator with this specification and header:

```
/** Return an Iterator over the elements of this list. */  
public Iterator iterator()
```
- H. Change the class of declaration of DLinkedList to implement Iterable<E>.
- I. If you did the above correctly, both classes should have no syntax errors. And if you Run class DLinkedListTest and made no mistakes in the above, the test should run without errors.
- J. Show your code to your TA, who will record that you finished this little exercise.