ICSI213 Data Structures

Notice that students are expected to start the lab as soon as the description is available and seek feedback during the lab. Labs are contiguous study of the lecture or used as stepping-stones for the projects. Skipping lab activities would impact the learning significantly.

Lab 04 *Iterator* Interface

For this lab, read the iterator PowerPoint(below) prior to the lab. Your co-instructor will study Iterators first. And then, you will write a Java program and submit it on Duifene on time. Javadoc style comments are required. Here are the instructions:

In *main*:

- Create an array list and use it to store a list of Integer objects.
- Create an iterator over the elements of the array list.
- Iterate through the list, print each object. Optionally, you can remove the object after it is printed.
- Check the size of the array list. If no objects were removed, the size should remain the same. Otherwise, the size should be changed.

main with sufficient Javadoc style comments	5 points
Total/ 5	5 points

Iterators

- An *iterator* is an object that is associated with a collection. The iterator provides methods for fetching the elements of the collection, one at a time, in some order.
- Iterators have a method for removing from the collection the last item fetched.

The Iterator Interface

- Iterators implement the `Iterator` interface. This interface specifies the following methods:
  - hasNext() : boolean
  - next() : E
  - remove() : void
- The `remove()` method is optional, so not all iterators have it.

2

Methods of the Iterator Interface

Method	Description
hasNext() : boolean	Returns true if there is at least one more element from the collection that can be returned, false otherwise.
next() : E	Returns the next element from the collection.
remove() : void	Removes from the collection the element returned by the last call to next(). This method can be called at least one time for each call to next().

3

The Iterator Interface

- A collection such as an array list has `iterator` method.

```
public Iterator<E> iterator()
```

- This method returns an iterator over the elements in this list in proper sequence.

4

5

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

/**
 * Demonstration of interface Iterator.
 * @author Tony Gaddis and Qi Wang
 * @version 1.0
 */
public class IteratorInterfaceDemo {
      /**
       * Creates an array list and displays the elements in order.
       * @param array The array to sort
       */
      public static void main(String[] args){
        List<String> names = new ArrayList<String>();

        //Add three elements into the array list.
        names.add("Anna");
        names.add("Bob");
        names.add("Carlos");

        // Get the iterator of the array list.
        Iterator<String>  it = names.iterator();

        // Iterate the list and display the elements.
        while (it.hasNext()){
            System.out.println(it.next());
            // The most recently returned element from next call is removed.
            it.remove();
        }

        //No more elements.
        System.out.println(it.hasNext());

        //The size of the array list is 0 if remove is called.
        System.out.println(names.size());
      }
}
```