This lab will be broken into two parts that are related:

1. Implementing sorting of a class.
    a. Adding sorting to a simple class using the **Comparable** interface
    b. Creating a custom sort using the **Comparator** interface

2. Implementing an iterator into a doubly linked list

Part I:

Create a class called Student with the following

- Properties:
    - fName, lName, age, test1, test2, test3, test4
- Constructors:
    - Student(String ln) – accepts a string representing a line from the given text file that will be parsed and have each property filled out.
- Methods
    - average(), fullName(), toString() – should print their fullName and average
- Additional:
    - By default the class should be sortable by the fullName() method.

Part II:

Download the files for Lab-06.

Modify the DoublyLinkedCollectionIteration class so that we can use a for each loop to iterate through the nodes

Tester Class

Create a Tester class and add the following two lists:

- DoublyLinkedCollectionIteration<Student> **dlStudents**
- ArrayList<Student> **alStudents**

1. Create a method in Tester called loadData() that will open the given text file and import all students into your program by creating a Student object for each row and adding it to both lists.

2. Create a method called getTotalAverage() that will iterate through your **dlStudents** and calculate the total average of all students.

3. Create a method called getTop10() that will sort the **alStudents** as they want to be sorted and prints them out.

4. Create a method called getBottom10() that will sort the **alStudents** in descending order based on their average