

Instructions: The exam is open book, open notes, and you may use a computer and Eclipse. You may communicate only with the instructor and the TA, Ching Yu, about the exam. This exam is due by 5:00 p.m. on Wednesday, 30 April 2014.

The maximum possible points that can be earned on the exam is as follows:

<u>Date Exam Submitted (both hard copy and electronic copy)</u>	<u>Max Points</u>
Tuesday, April 29, 2014	120
Wednesday, April 30, 2014	110
Thursday, May 1, 2014	100
Friday, May 2, 2014	90
Saturday, May 3, 2014	80
Sunday, May 4, 2014	70
Monday, May 5, 2014	60
Tuesday, May 6, 2014	50
Wednesday, May 7, 2014	40
After Wednesday, May 7, 2014	0

The following Java files are available for this exam: `DoubleLinkNode.java`, `DoubleLinkRing.java`, `DoubleLinkRing2.java`, `TestDoubleLinkRing.java`, and `TestDoubleLinkRing2.java`. This code can be found in the folder `/Users/cs64/Public/forExam2` on `ellafitz`. Notice that some of the lines of code are commented with the word "Picture" and an upper case letter or letters.

1.) Trace the execution of the code of the main method of class is `TestDoubleLinkRing` by drawing a picture of the data structure **and** each of the other variables **after** execution of each commented line of code or **after** execution of the if-else statement that is immediately before the Picture comment. The flow of execution may cause some lines of code to be executed several times. You need to draw a complete picture **each time** the line of code is executed. The flow of execution will **not** result in a set of pictures that is in alphabetical order. For example the first eight pictures to be drawn are Picture S, Picture A, Picture B, Picture C, Picture II, Picture JJ, Picture T, and Picture D.

Submit a hard copy of your pictures. [33 points]

2.) The code for manipulating a doubly linked ring can be greatly simplified if one uses a "dummy" node. In other words, an empty list contains one Node, the "dummy" DoubleLinkNode. A list with one real data item contains two DoubleLinkNodes, the "dummy" DoubleLinkNode and the DoubleLinkNode for the real data item. In general, a list containing n real data items contains $n+1$ DoubleLinkNodes because it has n DoubleLinkNodes for the data and one DoubleLinkNode for the dummy. The reference `myStart` always points to the "dummy" DoubleLinkNode. I have begun to write the class `DoubleLinkRing2` to take advantage of this strategy. That code can be found in the file `DoubleLinkRing2.java`. Note the change in the `add` method. The `add` method will not compile until you have written the code for the `search` method. You are to complete the rewriting of class `DoubleLinkRing2` to take advantage of the "dummy" DoubleLinkNode in simplifying and shortening the code wherever possible. The method skeletons have been commented out at this point as have some method invocations in class `TestDoubleLinkRing2`. When you have completed the code revision, execute the code with `TestDoubleLinkRing2.java` as the main class. It should produce the same output as the execution of the code with `TestDoubleLinkRing.java` as the main class. Submit both a hard copy and an electronic copy of your code to the Dropbox on `ellafitz`. [34 points]

3.) Trace the execution of the code of the main method of class `TestDoubleLinkRing2` by drawing a picture of the data structure **and** each of the other variables **after** execution of each commented line of code or **after** execution of the if-else statement that is immediately before the `Picture` comment. The flow of execution may cause some lines of code to be executed several times. You need to draw a complete picture **each time** the line of code is executed. The flow of execution will **not** result in a set of pictures that is in alphabetical order. Submit a hard copy of your pictures. [33 points]