

**PROJECT #3: 75 points – Due Sunday, September 23, by 11:59 p.m.**  
**CSE 274 – Fall 2018**

**Outcomes:**

- Implement an ADT (specifically a linked-node based implementation of the Set ADT in which all elements of the set are kept sorted)
- Test a class

**Naming requirements (not following any of these may result in a score of 0):**

- You have been provided two source code files that you must download and put in your project:
  - **SetInterface.java**, which should not be changed in ANY WAY.
  - **SortedSet.java**, which is where you will write most of your code
- In addition, you will write a third file, **SetTester.java**, which you will use to test your SortedSet (following the same guidelines as in project2) using the JUnit library.
- Use the **default package** (this means there should be no package statements in any of your files).
- The above three .java files should be the only source code files in your project.

Your assignment is to:

1. Implement the SortedSet class using linked nodes.
  - a. Create a private inner class named Node within the SortedSet class.
  - b. At all times, the set should be in sorted order. So, if a new item is inserted, or items are removed, it is important to make sure that the set is still sorted after these operations. **Do not use an array or any other Java data structures to help you sort. That would be very inefficient. You ought to be able to keep the data sorted without creating an extra array or list or map or table or an extra chain of linked nodes.**
2. The only instance variables should be a node representing the start, or head, of the nodes, and an int representing the size of the set.
3. There should be one constructor: A no-parameter constructor that creates an empty set.
4. SetTester should thoroughly test the methods and constructor in the Sorted class. It should utilize the JUnit library to test and comment on what is being tested, what results are expected. **This should not involve any interaction from the user.** Do not ask the user to enter input. Just run the test cases.

Your test code should not require me to look at your source code. I should know, by running your tester, what is being tested, what results are expected, and what the actual results are. I will be looking for:

- Thoroughness (test the constructor and all methods)
  - Organization (keep related tests together)
  - Readability (use single blank lines in appropriate places to break your code into "chunks" so it's easy to know when one part of your testing is done and the next part begins).
5. Submit your work (your three .java files) on Canvas.

### Grading Rubric:

Outcome	Max score
Constructors implemented correctly	2
Data is always kept sorted, without the use of arrays or extra chains of linked nodes	18
Remaining methods work as specified.	35
Tester is thorough, organized, and readable. Uses JUnit library	20
Code formatted according to generally accepted standards	0 (deductions only)
Code follows approaches taught in CSE 174 and 271	0 (deductions only)