**Assignment 6**

1. Modify class *SortedArrayCollection<T>* by introducing an instance variable “*comparator*” (*Comparator<T>* *comparator*), then follow what we did in class to modify the constructor and method “*add*” as necessary.
2. Write a class called *SortedArrayList<T>* that extends *SortedArrayCollection<T>* and implements interfaces *List<T>* and *Iterable<T>* with the following details:
3. To implement methods of *List<T>*, you may borrow some code we had in class if the code would work for sorted lists too. However, some implementations must be changed to maintain the order. In particular, methods *add(int index, T item)* and *set(int index, T item)* should do nothing if index is at a position where ‘item’ would destroy the order of the list.
4. Implement the following (new) method, which removes all, but one, copies of ‘*item*’ from the list:

*public void removeAllButOne(T item)*

1. Implement the iterator in such a way that iteration starts at the end of the list, i.e., the list is reversely traversed.
2. Practice on “Comparator”:
3. Write a class *Rectangle* with the following properties:
   * Two instance integer variables: *width* and *length*
   * Two methods: *getArea()* and *getPerimeter()*
   * Other methods to override: *toString* and *equals*
4. Then, create an instance of *SortedArrayList<T>* by providing an instance of *Comparator<Rectangle>* (you decide how to compare two rectangles), and then add a few objects of *Rectangle* into the list.
5. Display the list using Iterator methods. (The list should be displayed in reversal of the order you defined with Comparator.)

**Testing**:

Provide a test table ONLY for testing method *add(int index, T item)* in Question 2.