CS106/206 Assignment HC6/BMC7 Style Grading Rubrics

General

6 points are allocated to fairly mechanical rules on naming/comments/indentation - these should be easy to check off. Another **19** points are allocated to more creative practices, as explained below. Consult the formatting guide for details to check for under each category

Print student programs from Emacs, via “postscript print buffer” menu option.

Total: 25 points

Code formatting (**6 points total**)

1. Naming Conventions: **2 points**
    a. if any of the rules are violated
2. Whitespace: **1 point**
    a. inconsistent spacing (excessively) - - if just one place, point it out but don’t take off
3. Comments: **2 points**
    a. File header missing or malformatted
    b. Uncommented instance variables - no comment is okay if well-named
    c. Uncommented methods (getters and setters can have no comments, when appropriately named)
    d. Method comments that do not conform to javadoc style
    e. Uncommented complex blocks of code
    f. Unhelpful comments
4. Indentation: **1 point**
    a. inconsistent indentation (excessively) - if just one single line, point it out but don’t take off

Design principles (**19 points total**)

Assignment 7 (binaryheap)

1. private Instance variables and getters **1 point**
    a. Any non-private instance variables, including missing modifier
2. Use `public/private static final` constants instead of integer/double/String literals - any literal that has reason to be changed later should be a constant **1 point**
    a. Using `[0]`, `[1]`, `[2]` … directly in code after calling `split` (does not apply to Haverford, which uses `CSVReader`)
3. Constructor must initialize all instance variables **1 point**
4. `ArrayBinaryTree` (**5 points**)
    a. `Node` class is not required, but if they use one, it should be private and nested
    b. Any method that takes an array index parameter shouldn’t be public
    c. `remove` handles `null` by swapping with last element

    d. `toString*Order` and `toString` implemented

5. `ArrayHeap` (**9 points+EC**)
    a. `insert` and `remove` overridden with appropriate upheap and downheap operations
    b. `insert` and `remove` do comparisons via `.compareTo` for keeping heap order based on the keys (polling percentages)
    c. Object equality should be determined by `.equals` or `compareTo==0` based on last names
    d. `toStringBreadthFirstOrder` implemented
    e. `peekTopN` implemented and analysis in README is correct. Points are mostly for the analysis, as correctness checks functionality **4pts**
    Easiest just to copy into a second array and call Collections.sort on it. This is O(nlogn).
    If they manage anything better, which means they are able to separate the number of top items requested (k) and the total number of elements (n), give extra credit. This typically requires a partial sort or a second heap.
        i. **+1** O(1) if $k \leq 2$, just look in [0], [1], [2]
        ii. **+1** O(n+klogn) if copying into a second heap of size n and just poll first k
        iii. **+3** O(n+klogk) if copying into a second array and partial-sort the first k largest elements with quicksort. Only **+1** if using n^2 sort to partial sort, i.e. selection
        iv. **+5** O(klogk) if using a second heap in the following manner:

```
insert root of heap1 into heap2
repeat k times:
        poll heap2 and insert its two children (from heap1) into heap2
```

6. Custom class say `Candidate` for polling data (**2 points**)
    a. Implements `Comparable<Candidate>`
        i. if implements `Comparable` and `compareTo` casts `Object` to `Candidate` instead, don't take off but write comment that `Comparable<Candidate>` is better
    b. `.equals` overridden OR `compareTo == 0` is used to check equality
    c. `toString` overridden appropriately

Deductions:

7. Additional unnecessary data structures (take off 3-5 points depending on how bad it is)
8. More loops/calls than necessary - for example if `remove` calls `contains` to see if an item is in the tree, and then calls `containsIndex` to retrieve its index. -1 each