Name:_____

CS2852 Exam 2

No note-sheets, calculators, or other study aids on this exam. Write your name on all pages and read through the exam before you get started. The exam is printed double-sided.

Throughout the exam, write ***concisely*** and ***underline*** key words or phrases.

Have fun!

1. [20 points total] Consider the following expression "2 * (5 + 6)".

    a. (5 points) ***Draw*** a tree representation of this expression.

    b. (5 points) Following a post-order traversal, ***write a program-snippet*** that computes this expression. Instead of using elementary operators (*, +, etc.), use the following methods that are defined in "`this`" class.

    void push(int) – push an integer onto the stack
    void multiply() – pull two `int`s off the stack, multiply them, and push the result.
    void add() – pull two `int`s off the stack, add them, and push the result.

    c. (5 points) ***Illustrate*** what the stack will hold immediately ***before*** calling "***multiply***" (for the first time, if you use multiple calls).

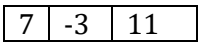
    d. (5 pts) ***Illustrate*** what the stack will hold immediately ***before*** calling "***add***" (for the first time, if you use multiple calls).

Name:_____

[30 pts total]

2. (5 points) You can look at the element about to come out of a Queue using either the element() or the peek() method. ***Explain*** the ***difference*** between these methods.

3. (10 points) ***Explain*** why an ArrayList<E> is not an appropriate choice when implementing a ***pure*** queue interface like the one we wrote in class.

4. (15 points) ***Illustrate*** a circular queue of capacity 10 that has had the following elements added to it in order: 7, -5, 0, 3, 8, 2, followed by removing 2 elements. Be sure that your illustration ***makes clear*** which element is at the front of the queue. You do not need to follow the full memory-map diagram used in class. You can show an array as, e.g. 

and an ArrayList as 

without needing to explicitly show references, the call stack, etc.

5. [25 points total]

   a. (20 points) ***Write*** a ***recursive*** implementation of an in-order traversal of a binary tree. Print the tree as you traverse it. You may assume the inner Node class has `left`, `right`, and `value` instance variables.

   [Can do pre- or post-order traversal and state this for nearly full credit.]

   b. (5 points) In Big-O Notation, ***write*** the order of everything in one call to your recursive method, ***excluding*** the Big-O cost of recursive calls to itself. In other words, you should count the cost of all operations in the method ***except*** for the lines where the recursive algorithm calls itself. ***Explain*** your answer.

Name:_____

[25 pts total]

6. (5 points) ***Draw*** a binary tree that is ***complete*** but ***not full***.

7. (5 points) ***Explain*** why you would use the Java API `Queue` methods that throw exceptions if you wanted to store a "null" in the `Queue`.

8. (5 points) ***Explain*** how you could write your code to use the `remove()` method in the Queue interface, not catch the exceptions thrown by it, ***and yet not crash***.

9. (5 points) ***Explain*** why the binary search algorithm for a sorted array requires significantly fewer comparisons in the worst case than the search algorithm for an unsorted array.

10. (5 points) Is a queue or a stack LIFO? ***Explain*** your answer.