Midterm Exam 600.226 Data Structures Example Questions

Name: _____ Email: _____

This is a closed-book, in-class exam. Answers must be your own. There is no penalty for minor Java syntax errors, or slightly misnamed methods, if it's clear you have the right idea. Explaining your answer will help us give you partial credit if it's wrong.

Section I: Short Answer (2 pts each, 14 pts total)

1. What type of Java variable is shared among all instances of a class?

2. Define the term *polymorphism*.

3. What is worse, $O(n \log(n))$ or $O(n^{3/2})$?

4. What data structure implements FIFO behavior?

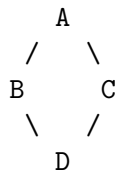
Section II: Medium Answer (33 pts total)

1. Using $<$ and $=$ order the following functions by asymptotic growth rate:

   (a) $4n\log(n) + \sqrt{n}$
   (b) $2^{\log(n^2)}$
   (c) $3n + 20\log(n)^2$
   (d) $\log(n^5)$
   (e) $n\log(n)$
   (f) 2^{200}

   Answer: f ¡ $b = d$ ¡ a ¡ c ¡ e

2. (6 pts) Consider the following inheritance diagram for A, B, C, and D:

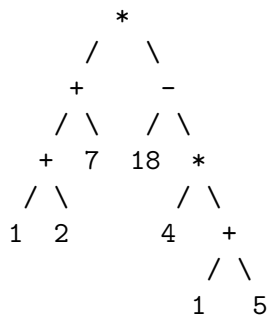


   List 3 ways that this diagram could be implemented in Java by interfaces (I), abstract classes (A), or classes (C).

	1	2	3
A			
B			
C			
D			

Section III: Long Answer (28 pts total)

1. (16 pts) Consider the following expression tree



(a) What is the formula represented by the tree?

(b) What is the sequence of symbols that would result from a postorder traversal?

(c) Suppose I want to apply the associative law on the left branch and group the addition of 7 and 2 before adding 1. What sequence of operations from the LinkedBinaryTree class would accomplish this?

2. (12 pts) *Use the back of one of the other test papers to answer this question* Suppose you are asked to use two stacks, RedStack and BlueStack, as your only instance variables to implement the Queue abstract data type.

   (a) Describe in pseudo-code how you would implement the methods enqueue() and dequeue().

   (b) Use the big-Oh notation to characterize the running times for the methods enqueue() and dequeue() in this implementation in terms of n, the number of elements in the queue.