Do Not Turn This Page Until Directed to do so.

Name: ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

You have 75 Minutes minutes from the start of class to complete this exam.

If you fail to turn in your test before leaving the room, check your cell phone, or you use any resource other than your card of notes, your test will be taken and you will be **given a grade of 0**.

Your note card must be submitted with the test.

Read the instructions carefully and answer each question as completely as you can.

There will be no talking during the test. This also means that I will not answer questions. Do what makes sense for each question.

Question	Points	Score
1	4	
2	4	
3	4	
4	4	
5	4	
6	4	
7	4	
8	4	
9	4	
10	4	
11	4	
12	4	
13	4	

Question	Points	Score
14	4	
15	4	
16	4	
17	4	
18	4	
19	4	
20	4	
21	4	
22	4	
23	4	
24	4	
25	4	
Total:	100	

Points: _____

This page is left blank to use as scratch paper.

1. (4 points) Suppose you have a stack (initially empty) with standard push(int e), pop(), and peek() methods. What are the values in the stack after the following operations were executed: pop(), push(5), push(3), peek(), push(7), pop(), push(2), pop(), pop()

   A. null
   B. 2
   C. 5
   D. 3, 5

2. (4 points) Which of the following is a true statement about interfaces in Java?

   A. They enable the structuring of classes into a hierarchy.
   B. They provide a list of methods guaranteed to be in any class that implements it.
   C. They process ActionEvents to enable MouseListeners and GUI interactivity.
   D. They are instantiated to provide flexibility with run-time object selection.

3. (4 points) What is the reason for using Generic Classes when building Classes?

   A. Generic Classes avoid stack overflow errors on recursive calls.
   B. Generic Classes enable operations to occur for different data types.
   C. They help ensure code will function on one single data type.
   D. They identify and enforce specific execution parameters.

4. (4 points) What is a major limitation of using an array to implement a stack?

   A. Node creation is complicated because they will be stored inside the array.
   B. Directly indexing into the array will slow operations when searching a stack.
   C. Insertions into and deletions from the array need to involve a lot of shifting.
   D. Either the array needs to be resized, or the stack needs to have a fixed size.

5. (4 points) If you can possibly use recursion, you should. I.e., recursive solutions are ALWAYS better than iterative solutions.

   A. TRUE
   B. FALSE

6. (4 points) How should one fix a NullPointerException?

   A. Go to Isaac, he'll fix it.
   B. Go to the line with the error and look at the method that is being called in that line.
   C. Go to the line with the error and identify the object that doesn't exist but should.
   D. Go to the line with the error and see if the recursive call is structured properly.

7. (4 points) What is likely happening when you get a StackOverflowException?

   A. You probably have a recursive function that is not stopping.
   B. You probably are accessing a linked list without checking if head == null.
   C. You probably missed creating a new instance of whatever class threw the exception.
   D. You probably have a variable that was never initialized.

8. (4 points) I am building an app that tracks the weather in Bozeman. People will be delivered the current day's weather. They may want to go back and forward a couple days in case they didn't check it recently. They may also want to check the weather at arbitrary days in the past. What should I use to store my weather data?

    A. Circularly Linked List

    B. Doubly Linked List

    C. Linked List

    D. Array

9. (4 points) My company is subject to government oversight. As such, I need to keep records detailing the internal financial review I do every month. I want to store them in chronological order (this month, then last month, then the month before, …). It is very, very unlikely that I will be audited, so I am not concerned about needing to search through my records very often. What should I use to store my records?

    A. Circularly Linked List

    B. Doubly Linked List

    C. Linked List

    D. Array

10. (4 points) I am climbing a wicked new route on some remote Alaskan peak. Being the weight weenies we are, my partners and I only brought one spoon (the handle is also cut off to shave grams). Not only does this mean that we need to share the spoon, but more importantly it means we need to take turns digging into the instant mashed potatoes. We want to be fair about making sure everyone gets their turn, so what would be best to use to keep track of whose turn it is to eat?

    A. Circularly Linked List

    B. Doubly Linked List

    C. Linked List

    D. Array

11. (4 points) I am building an application that tracks patients for the emergency room. The person at the entrance will enter patients into the system and the doctor in the back will pull from the system (and see) whoever is closest to death. What data structure should I use to store the patients in my system?

    A. Deque

    B. Stack

    C. Queue

    D. Priority Queue

12. (4 points) I am packing 15 clowns into a small car (I don't ask what you do in your personal time). There is only one door and the plan is to pack them all in, then drive somewhere and unload them all. What data structure should I use to keep track of the order clowns enter and leave the car?

    A. Deque

    B. Stack

    C. Queue

    D. Priority Queue

13. (4 points) Rumor has it that the iPhone 7.3s 9G XYZ is about to hit stores. Apple has noticed in the past that their customers are an odd bunch. They will wait in lines until the end of time, unless they are the last person in the line, then they may leave the line. Clearly people will leave from the front also as they are granted the privilege of handing over $500. What should Apple use to model the lines?

A. Deque

B. Stack

C. Queue

D. Priority Queue

14. (4 points) I have decided to follow my dream of becoming a TV Chef. Everyone knows that this means that I need to have all my ingredients prepped before I start cooking. The way I do that is to go through the recipe one line at a time (in order), and prep the ingredients as I will add them (chop the onion, then measure the chicken stock,…). After that is done, I will then roll the camera and use the ingredients in the order they were prepared. What sort of data structure should I use to represent my line of ingredients?

A. Deque

B. Stack

C. Queue

D. Priority Queue

15. (4 points) What most accurately describes a linked list?

A. An array of Nodes that contain values. They can be visited in order by going through the array.

B. Entities that consist of pointers to other entities in a circular fashion.

C. A Node that has some sort of data payload as well as methods that let you insert and remove that data.

D. A sequence of objects containing a value and pointer to the next object, as well as a pointer to the first object.

16. (4 points) How could you implement a priority queue using what you know now?

A. Use a stack. Push low priority items first so they are at the bottom, the popping will remove the top (high priority).

B. Create an object with value and priority parameters, then use a regular linked list. Insert in order to enqueue and remove from the front to dequeue.

C. Use a doubly linked list, but enqueue by adding to the front and dequeue by removing from the rear. Priorities can be ignored because it is doubly linked.

D. Use a regular ArrayList implementation of a queue. The priority will be then correspond to the spot in the array to place the value (i.e. a priority 1 item would could into spot 1).

17. (4 points) Suppose you wanted to simulate how long it takes a mailman to deliver all the letters in his truck. Suppose he delivers a random number of letters per minute (between 3 and 7). What would be the most reasonable approach?

A. Set up a simple program that will register the System.nanoTime() each time you hit enter. Fill a mailman's truck up with the correct number of letters. Hit enter, follow him around until he is finished (ensuring his delivery rate is between 3 and 7 per minute) and hit enter. The time it took is the difference of those two values.

B. Use a while loop to keep running while the time used is below some threshold. Inside the while loop, deduct an appropriate random number of letters from the truck. If the truck empties before the time is up, rerun with a smaller threshold. If the truck is full after the time is up, rerun with a larger threshold.

C. Use a while loop to keep running while the truck isn't empty. Inside the while loop get an appropriate random number of letters and deduct that from the letters in the truck. Return the number of times the while loop executed (number of minutes it took).

D. Use a for loop to keep track of the number of minutes used. Inside the for loop, if the truck isn't empty, remove an appropriate random number of letters from the truck. When the time is expired, return the number of letters left in the truck.

18. (4 points) Which growth function would be best for an algorithm to have for very large values of n?

    A. $f(n) = 100 * n^{2.2}$

    B. $f(n) = log(n) * n^2$

    C. $f(n) = .5 * n^2$

    D. $f(n) = 1.001^n$

19. (4 points) What is an appealing property special to linked lists (when compared to arrays)?

    A. Swapping nodes is easier than swapping spots in an array.

    B. Searching through them is fast.

    C. It is easy to determine what the next value is.

    D. Easy to resize.

20. (4 points) Linear search can work on unsorted data, but binary search must be done on sorted data.

    A. TRUE

    B. FALSE

21. (4 points) Suppose I had a queue that was initially empty and I performed a total of 18 offer operations, 7 first operations, and 4 poll operations. Suppose 2 of the poll operations and 3 of the first operations returned null values. How many items are in the queue?

    A. 0

    B. 13

    C. 14

    D. 16

22. (4 points) For the following method, identify the tightest big-oh running time.

```
public int m1FindLargest(int[] array) {
    if (array.length != 0) {
        int value = array[0];
        for (int i = 1; i < array.length; i++) {
            if (array[i] > value) {
                value = array[i];
            }
        }
        return value;
    }
    return -1;
}
```
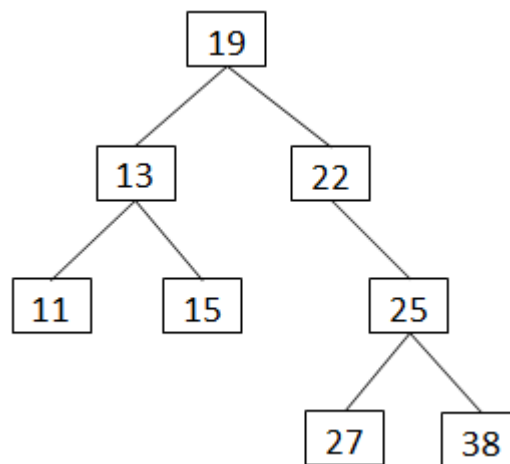
    A. $O(n!)$

    B. $O(n)$

    C. $O(\log n)$

    D. $O(n^2)$

23. (4 points) What is the worst case running time to search for a value in a binary tree?

    A. $O(n^2)$

    B. $O(1)$

    C. $O(n)$

    D. $O(\log n)$

24. (4 points) Suppose I have a binary tree that looks like the image below. What is the correct ordering of values returned with a post order traversal of this binary tree?

    A. 11, 15, 13, 38, 27, 25, 22, 19

    B. 19, 13, 11, 15, 22, 25, 27, 38

    C. 11, 13, 15, 19, 22, 27, 25, 38

    D. 11, 15, 13, 27, 38, 25, 22, 19

25. (4 points) Which of the following is true of selection sort?

    A. It runs surprisingly fast on large arrays.

    B. As soon as an element is placed, it is in its final spot.

    C. It does not require any local variables to sort.

    D. It does not need to wait for all the data to be collected before it can begin sorting.

End of Test