CSCI 2320
Spring 2018
Exam 3

Last Name: _____

First Name: _____

Write your answers on the answer sheets provided. You may refer to printouts of any source code you have written.

```
class PetSnake {
   private String name;
   private int length;  // in cm
   public PetSnake(String name, int length) {
      name= name;     // this ain't right!
      length= length;
   }
}
class ListNode {
   double val;
   ListNode next;
}
class MyList {
   private ListNode head;  // head of the linked list
   ...
}
```

1. (4 pts) Expand the acronym LIFO. What data structure is it associated with?
2. (2 pts each) Briefly define each of the following terms (in the context of this course):
   (a) queue
   (b) `.compareTo()`
3. Answer these questions about the `PetSnake` class listed above.
   (a) (4 pts) The constructor, as written, does not function as we would want. Show how to fix it.
   (b) (6 pts) Write a `.compareTo()` method for the `PetSnake` class that will declare one snake "less than" another snake if it is shorter. If two snakes are the same length the method should declare the snake whose names appears earlier in the alphabet to be "less than" the other snake. If the snakes have the same length and same name then they are considered equal.
   (c) (4 pts) Write a `.toString()` method that will return the snake's name.
   (d) (6 pts) Assume there are properly written `.compareTo()` and `toString()` methods as described above. Write a section of code that will create two `PetSnake` objects (Fred / 20cm; Alice / 43cm). Then compare them (using `.compareTo()`) and, in your code, inspect the result. Provide logic so that depending on the comparison it will print one of these messages:

- (name of snake 1) is less than (name of snake 2)
- (name of snake 1) is greater than (name of snake 2)
- (name of snake 1) is equal to (name of snake 2)

(e) (2 pts) What would be output by the code you wrote in the previous problem?

4. Answer these questions about a linked list based on the `MyList` class above.

   (a) (8 pts) Write a method for the `MyList` class called `toArray()` that will copy the data from the list into an array. The method should return a reference to the array. As a first step, traverse the list to count the number of elements and then create an array of that size. Then traverse the list again to copy data into the array.

   (b) (8 pts) Suppose the list in `MyList` is ordered. Write a `fancyDisplay()` method that will display all elements in the list to the screen with one entry per line. If, however, a particular value appears more than once in the list it should only be printed once with the text "(x times)" where x is the number of times the value appears in the list.

   (c) (8 pts) Write a method for `MyList` called `reverse()` that will reverse the order of elements in the list. Do not reserve any additional memory as part of your solution.

   (d) (4 pts) What changes would you make to the `MyList` class and the `ListNode` class to make it a generic list?

5. (6 pts) Show how the following array would be processed if it were partitioned one time using the partition algorithm that accompanies Quicksort.

   $< 75, 86, 151, 51, 143, 81, 94, 48, 61, 91, 111 >$

6. (4 pts) Write pseudocode for a binary search.

7. (4 pts) Suppose you peform a binary search on an array containing 100 elements. How many comparisons would it take to determine that an element being searched for is not there?

8. Answer these questions about stacks and the `MyStack` class given above.

   (a) (4 pts) Suppose that when a letter is encountered we push it on the stack and when an asterisk is encountered we pop the stack. Show the sequence in which letters would be popped for the following string:

   ```
   EON**REEM**GS*AS***E*H****
   ```

   (b) (4 pts) If you were called upon to create a custom-built stack would you utilize an array or a list. Explain the reason for your choice.

9. Answer these questions about queues.

(a) (4 pts) Suppose that when a letter is encountered we put it in the queue and when an asterisk is encountered we remove from the queue. Show the sequence in which letters would be removed for the following string:

```
EON**REEM**GS*AS***E*H****
```

(b) (8 pts) Suppose you are going to use Java's built-in `LinkedList` class as a queue. This can be accomplished by using the `.add()` method to insert elements into the list and using `.remove()` to remove elements from the list. You will get queue-like behavior because those two methods work on opposite ends of the list! Consider this declaration:

```
LinkedList<String> queue= new LinkedList<String>();
```

Suppose that the variable `queue` above has been filled with hundreds of names that represent people who are wanting to enter a concert venue. It has been determined that entry will be granted initially to every 5th person (in order). Once those people have been admitted then the remaining people will be invited using the same technique (every 5th person). This will continue until everyone has been named. Use the variable `queue` and only its `add()`, `remove()`, and `size()` methods to display the order in which people will be admitted. NOTE: this is a simpler (and kinder) version of the Josephus problem.

10. (4 pts) Which homework assignment in this course was your favorite? Why?

11. (4 pts) Which topic in this course was your favorite?