CSCI 2320 Last Name: ____________________
Spring 2025
Exam 3 First Name: ____________________

Write your answers on the answer sheets provided. You may refer to printouts of the Java Command Sheets (without commentary) provided in class.

```
class ListNode {
   int data
   ListNode next;
}
class MyList {
   public ListNode head;
   public MyList() { head= null; }
   public void add(int elem) { ... /* adds to front of list */ }
   public void remove(int elem) { ... /* removes specified element list */ }
}
class MyQueue {
   // internal implementation not specified, but does have these methods
   public void insert(double elem) {}
   public double remove() {}
   public int size() {}
}
```

1. Answer these questions about binary search:
   (a) (3 pts) Why can't binary search be applied to every array?
   (b) (2 pts) About how many steps/iterations would it take a binary search to find an element in a list of 1000 elements.

2. (6 pts) Show how the following array would be processed if it were partitioned one time using the partition algorithm that accompanies Quicksort.

   $< 44, 55, 120, 20, 112, 50, 63, 17, 30, 60, 80 >$

3. Consider this `Customer` class used to represent cable customers:

```
class Customer {
   private String email;            // customer email address
   private int cablePackage;        // 1=basic; 2=supreme; 3=everything
   private boolean wantsPromotions; // false= don't send promotional emails
}
```

   (a) (6 pts) Write a well-formed Java compareTo() method for this class so that a customer is considered less than another customer if their email address is lexicographically less than the other customer's email address.

(b) (4 pts) Show how the quicksort partition method below would be changed if it were to be used to sort Customer objects rather than integers. Write your answer on this exam sheet by writing in the changes rather than by rewriting the the entire method.

```
public static int partition(int [] wordList, int start, int stop) {
   int left, right;
   int temp;
   int k= wordList[start];

   left= start-1;
   right= stop+1;

   do {
      do { right--; } while (wordList[right] > k);
      do { left++; } while (wordList[left] < k);

      if (left < right)
      {
         temp= wordList[left];
         wordList[left]= wordList[right];
         wordList[right]= temp;
      }
   } while (left < right);
   return right;
}
```

4. (4 pts) Expand the acronym LIFO. What data structure is it associated with?

5. Answer these questions about linked lists and the `MyList` class given above.

   (a) (6 pts) Write the body of the `add` method in the `MyList` class.

   (b) (6 pts) Write a `count(int elem)` method for the `MyList` class that will return the number of elements in the `MyList` object that match the parameter `elem`.

   (c) (8 pts) For this problem assume the values in `MyList` are in ascending order. Write a `printDuplicates()` method for the `MyList` that will display a list of duplicated values. If a value appears 2 or more times it should only be printed once. Other values should not be printed.

   (d) (8 pts) Write a `merge(MyList other)` method for the `MyList` that will combine the contents of the current list with the list specified by the parameter `other`. The current list should be modified but the `other` list should not. When combining list elements interleave them starting with the current list. For example:

Before merge:	**After list1.merge(list2):**
list1 = a, b, c, d, e	list1 = a, 1, b, 2, c, 3, d, e
list2 = 1, 2, 3	list2 = 1, 2, 3

6. 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:

   ```
   ST*ELEH**EBOL***TS***UR*R*****
   ```

   (b) (4 pts) Contrast an array-based stack implementation with a list-based stack implementation. Give advantages of each over the other.

7. 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:

   ```
   ST*ELEH**EBOL***TS***UR*R*****
   ```

   (b) (10 pts) Assume you are given a working `MyQueue` class implemented with only 3 methods (insert, remove, size). Write a method called `same()` that accepts two `MyQueue` objects as parameters and returns true if the queues contain the same elements (in the same order) as each other (false otherwise). When the method completes the queues should be unharmed. You may not modify the `MyQueue` class and you may only use the three provided methods when interacting with the queues. HINT: It will be helpful to first check the sizes of the two queues. If they are different sizes there is no need for further checking.

8. (4 pts) Which homework assignment in this course was your favorite? Why?

9. (4 pts) Which topic in this course was your favorite?