CMSC 132 Quiz 2 Worksheet

The next quiz for the course will be on Wed, Feb 27. The following list provides more information about the quiz:

• The quiz will be a written quiz (no computer).
• The quiz will be in lab/discussion session.
• Closed book, closed notes quiz.
• Answers must be neat and legible. You must use pencil.
• Check the information available at http://www.cs.umd.edu/~nelson/classes/utilities/examRules.html

The following exercises cover the material to be included in this quiz. Solutions to these exercises will not be provided, but you are welcome to discuss your solutions with the TA or instructor during office hours.

Exercises

1. The TA interface defines a single method with the following signature: public void assist(); Complete the following assignment where x is assigned an object that implements the TA interface and the method assist() will print (using System.out.println) the message "Holding office hours".

   TA x =

2. Give the asymptotic bound of the following functions:
   a. \( n^3 + \log(n) \)  \( f(n) = O(\quad) \)
   b. \( n^2 - 200n - n^2 \)  \( f(n) = O(\quad) \)
   c. \( 2n^4 - 500n - n^2 \)  \( f(n) = O(\quad) \)

3. List the following big-O expressions in order of asymptotic complexity (lowest complexity first).
   \( O(n \log(n)) \)  \( O(\log(n)) \)  \( O(n^2) \)  \( O(n^3) \)

4. What is the complexity of computing \( a[i] \) for an array of length \( n \)?

5. Make the following class generic, so that it can deal with an arbitrary class rather than only Strings.

   public class Col {
       private ArrayList<String> c;
       
       public String get() { return c.remove(0); }
       public void insert(String value) { c.add(value); }
   }

6. Complete the following variable declaration so we have an ArrayList with objects that implement the Comparable interface.

   ArrayList< > aList;

7. Complete the following variable declaration so we have an ArrayList with objects that belong to the Reptile class or are subtypes of this class.

   ArrayList< > bList;
8. A Bag is similar to a set, except it we can keep track of how many instances have been added for a particular element. For this problem:

   a. Implement a Bag class that keeps track of strings added to the bag.
   b. The following methods should be supported:
      
      i. void add(String elem) → adds element to the bag.
      ii. int getCount(String elem) → returns the number of instances associated with elem.
      iii. String toString() → prints each string and number of instances for that string.
      iv. Define an iterator for the bag that allow us to go through all instances of the bag. For example, if the bag has two instances of the string “cat”, it will take two calls to next() in order for us to retrieve everything the bag has. You must use inner classes for the iterator.
      v. Define the remove method of the iterator interface. The method will reduce the number of instances of a particular string by one.
      vi. Feel free to add any other methods/instance variables you understand are necessary.