

CMSC 132 Quiz 3 Worksheet

The third quiz for the course will be on Wed, Nov. 4. The following list provides additional information about the quiz:

- The quiz will be posted on Wed, Nov. 4, 9 AM (morning), and due the same day, Wed, Nov. 4, at 5 PM (afternoon).
- It is designed to be completed in less than 1 hour, but I am making it available for 8 hours since people have different schedules.
- You will not have lab on Wed. Nov. 4, so that should free up 1 hour for everyone to work on the quiz. The lecture video for that day will be posted before 8AM, so students in the 8AM lab can use the lab time for lecture and the lecture time for the quiz.
- We will have normal office hours on Wed. Nov. 4, but TAs cannot answer any questions about the quiz in OH (They can help you submit if you have submit server issues).
- Did you install the correct version of Eclipse, Java 13, and course management software on your computer at the start of the semester? See here: <http://www.cs.umd.edu/eclipse/install.html>

If you don't have this exact setup and you are not able to submit the quiz, that will not be a valid reason for an extension.

- The quiz will be posted similar to a class project. You will write code in an Eclipse project and submit as usual.
- You can only post clarification questions in Piazza on quiz day and a CMSC 132 staff member will reply. **As a student, you should not reply to questions posted by other students about the quiz.** Debugging questions, why code is not compiling, why is code not passing a test, are invalid questions to post in Piazza.
- Posting of any kind of code in Piazza (or other public platforms), during the quiz period, represents an academic integrity violation and will be reported as such.
- The quiz will be graded based on submit server tests (release and secret) and code inspection (e.g. style, following rules, etc.). The exact rubric will not be available before the quiz. Just follow all the rules to avoid point deductions
- **You must work by yourself.** Sharing of quiz solutions represents an academic integrity violation and will be reported as such. Submissions can be checked with cheating detection software.
- You can use class resources (lecture notes, lecture/lab examples, videos, etc.), but no other resources (e.g., code from the web).
- All submissions must be done via the submit server (no e-mail). The highest scoring submission will be used for grading purposes (you can submit as many times as you want before the deadline).
- There will be a 1-hour late submission period, therefore you need to submit often and before Wed, Nov 4, at 5 PM (afternoon) for your quiz to count on time. If you turn it in between 5 and 6 PM, it will be marked late and there will be a 5-point deduction. Questions will not be answered on piazza during the late period.
- If you are student with an extended time accommodation from ADS, the time frame provided takes into consideration your time allocation. If you need any other assistance or still have concerns to finish the quiz, contact me via email before the quiz day.
- It is in your best interest to complete this work by yourself, and following the guidelines provided above. You need to identify which topics you understand and which ones you don't, so you can be successful in CMSC216 and future CS courses. The following exercises gives you practice with concepts that may show up on the quiz. Solutions to these exercises will not be provided, but you are welcome to discuss your solutions with the TAs during office hours or on Piazza.

Exercises

1. What is the Java Hash Code Contract?

2. Implement the methods below based on the following Java class definitions. For recursive methods you may only add one auxiliary function, and you may not add any instance nor static variables.

```
public class LinkedList<T extends Comparable<T>> {
    private class Node {
        private T data;
        private Node next;

        private Node(T data) {
            this.data = data;
            next = null;
        }
    }

    private Node head;

    public LinkedList() {
        head = null;
    }
}
```

- a. Define a constructor that takes a **TreeSet<T>** as a parameter and initializes a linked list with the elements in the set. The new list must be sorted in increasing lexicographic order.
- b. Define a **RECURSIVE** method named **size** that returns the number of elements in the list. The prototype for this method is:

```
public int size()
```

- c. Define a **RECURSIVE** method named **inRange** that returns a **HashSet<T>** with the elements in the list that in the specified range. The range includes the lower and upper bound. The prototype for this method is:

```
public HashSet<T> inRange(T lowerBound, T upperBound)
```

- d. Define a **RECURSIVE** method named **remove** that removes all instances in the list that are equal to the target parameter. The prototype for this method is:

```
public void remove(T target)
```

- e. Define a **RECURSIVE** method named **positionOfElementInList** that returns a **TreeMap<T, Integer>** that maps each element of the list to its position in the list. The prototype for this method is:

```
public TreeMap<T, Integer> positionOfElementInList()
```

- f. Define an equals method for the class. Two lists are equal if the data elements in the two lists are the same (and appear in the same position). Make sure your class satisfies the Java Hash Code Contract.

3. Below are actual quizzes from Fall 2018 and Fall 2019. I am just making this available for further practice, but remember the format of your quiz in Fall 2020 will be different than what we did in Fall 2018 and Fall 2019.

- <http://www.cs.umd.edu/class/fall2020/cmsc132/quizzes/Quiz3Fall2018.pdf>
- <http://www.cs.umd.edu/class/fall2020/cmsc132/quizzes/Quiz3Fall2018Soln.pdf>
- <http://www.cs.umd.edu/class/fall2020/cmsc132/quizzes/Quiz3Fall2019.pdf>
- <http://www.cs.umd.edu/class/fall2020/cmsc132/quizzes/Quiz3Fall2019Soln.pdf>