

## **CMSC 132 Quiz 1 Worksheet**

The first quiz for the course will be on Wed, Sep 16. The following list provides additional information about the quiz:

- The quiz will be posted on Wed, Sep 16, 9 AM (morning), and due the same day, Wed, Sep 16, 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. Sep 16, so that should free up 1 hour for everyone to work on the quiz.
- We will have normal office hours on Wed. Sep 16, 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, 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 (style). The exact rubric will not be available before the quiz.

• **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 is no late submission period, therefore you need to submit often and before Wed, Sep 16, at 5 PM (afternoon). Don't wait till the last minute to submit. If you wait till the last minute and there is a network delay, it will be a zero. I would recommend you finish and submit no later than 4:30 PM.
- 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, contact me via email.
- 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. Define a class Telephone according to the following information:

Instance Variables (all private)

- a. area code → integer value
- b. three digit value → integer value
- c. four digit value → integer value
- d. user name → String reference

### Instance Methods

- a. **Constructor** - Allows you to initialize all the instance variables of the class. Name the parameters after the instance variables (i.e., you must use the **this** reference)
- b. **Default constructor** - Initializes the object to the number 555-555-5555 and the name to null. This constructor relies on the previous constructor for the object initialization (i.e., you must use the **this** reference)
- c. **Copy constructor**
- d. **Get/Set methods** - Define get/set methods for all instance variables of the class.
- e. **equals** - Two numbers are considered the same if they have the same area code, three and four digit values. Use instanceof to implement this method (the approach described in class).
- f. **toString** - Returns a string with the user name followed by the phone number of the person.

### Static Variable (private)

- a. **count** - keeps track of how many Telephone objects has been created.

### Static Method

- a. **getCount** - Returns the count value
- b. **getDigits** - Takes a String reference as a parameter and returns the number associated with the string. For example, if the String has the value "CAR" the method will return the integer 227

2. What is encapsulation? How does it relate to abstraction?
3. What is the difference between procedural abstraction and data abstraction?
4. Do you know examples of academic integrity violations? The following is the list you need to know:
  - a. Hardcoding of results in a project assignment.
  - b. Using any code available on the internet/web.
  - c. Hiring any online service to complete an assignment for you.
  - d. Posting your implementation of any class project on the internet/web.
  - e. Discussing projects with your classmates.
  - f. Sharing your code or your student tests with any student.
  - g. Looking at another student's code.
5. Define an enumerate type named **Day** that represents the days of the week. Using the enhanced for loop construct, write a code fragment that prints all the days of the week.
6. The **PrinterJob** class is defined as follows:

```
public class PrinterJob {
    private int id;
    private int size;

    public PrinterJob(int id, int size) {
        this.id = id;
        this.size = size;
    }

    public int getId() { return id; }

    public int getSize() { return size; }

    public String toString() {
        return "Id: " + id + " Size: " + size;
    }

    public int hashCode() {
        return id;
    }
}
```

- a. Add a new private field "jobType" as an enumerated type with the values COLOR and BW.
  - b. Modify the class so it implements the Comparable interface, allowing you to compare PrinterJob objects based only on their id.
7. Review interfaces and understand that an interface creates a new type that can be assigned objects of the class that implement the interface (**is a** relationship).
  8. Review the for each loop (enhanced for loop).
  9. 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.

- a. <http://www.cs.umd.edu/class/fall2020/cmsc132/quizzes/Quiz1Fall2018.pdf>
- b. <http://www.cs.umd.edu/class/fall2020/cmsc132/quizzes/Quiz1Fall2018Soln.pdf>
- c. <http://www.cs.umd.edu/class/fall2020/cmsc132/quizzes/Quiz1Fall2019.pdf>
- d. <http://www.cs.umd.edu/class/fall2020/cmsc132/quizzes/Quiz1Fall2019Soln.pdf>