# CMSC 131 Quiz 3 Worksheet

The next quiz for the course will be on Wed, Apr 8. The following list provides additional information about the quiz.

- The quiz will be posted on Wed, Apr 8, 9 AM (morning), and due the same day, Wed, Apr 8, at 5 PM (afternoon).
- 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. 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.**
- There is no late submission period, therefore you need to submit often and before Wed, Apr 8, at 5 PM (afternoon).
- The quiz will be graded based on submit server tests (release and secret) and code inspection (style).
- You must work by yourself.
- 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).
- Sharing of quiz solutions represents an academic integrity violation and will be reported as such.
- Submissions will be checked with cheating detection software.
- Do not wait until close to 5 PM to submit your work. Network problems or submit server overloading are not valid excuses for missing a quiz submission. Plan to submit before 5 PM.
- If you are an ADS student: The time frame provided takes into consideration your time allocation. If you need any other assistance, contact your instructor. Ignore this entry if you don't know what an ADS is.
- **Dear Students: 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 CMSC132.**

**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

## 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's 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. Define the equals methods as described in class examples.
   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 Methods

a. **getCount** – Returns the count value.
b. **getDigits** – If you look at your phone's numeric keypad, you will see letters underneath the numbers (e.g., for 2 you will see ABC). The getDigits method takes a string reference as a parameter, and returns the number associated with the string by mapping each character of the string to a number. For example, if the string has the value "CAR" the method will return the integer 227.

2. What is an exception?

3. Modify the Telephone constructor so the exception IllegalArgumentException is thrown if the user's name is null.

4. Define a try catch block that handles the IllegalArgumentException when creating a Telephone object. The catch clause should print the message "Invalid argument" when the exception takes place. You can assume the values used to create a Telephone are provided by the user (using the Scanner class).

5. Previous Quiz

http://www.cs.umd.edu/class/spring2020/cmsc131/quizzes/Quiz1_133_Spring2020.pdf
http://www.cs.umd.edu/class/spring2020/cmsc131/quizzes/Quiz1_133_Spring2020_Soln.pdf