

CMSC 132 Quiz 1 Worksheet

The first quiz for the course will be on Thu, Jun 12. The following list provides more information about the quiz:

- The quiz will be a written quiz (no computer).
- 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. Read the document available at:

<http://www.cs.umd.edu/~nelson/documents/SuggestionsForWritingComputerPrograms.htm>

2. 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.
- 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

3. What is encapsulation? How does it relate to abstraction?
4. What is the difference between procedural abstraction and data abstraction?

5. Write a static method that is given a 2-dimensional array of doubles, and finds the first row that consists of an increasing sequence of values, that is, it finds the smallest i such that:

$$\text{array}[i][0] < \text{array}[i][1] < \text{array}[i][2] < \dots$$

If such a row exists, a copy of the contents of that row is returned as a one dimensional array. Otherwise null is returned. The signature of the method is:

public static double[] firstIncreasing(double[][] array);

6. 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.
7. 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;
    }
}
```

- Add a new private field "jobType" as an enumerated type with the values COLOR and BW.
- Modify the class so it implements the Comparable interface, allowing you to compare PrinterJob objects based only on their id.
- Implement a comparator class named SizeComparator that allow us to compare PrinterJob objects based on their size.