

CMSC 132 Quiz 1 Worksheet

The first quiz for the course will be on Fri, Jun 13. 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. We recommend that you use pencil and eraser.

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. What is encapsulation? How does it relate to abstraction?
2. What is the difference between procedural abstraction and data abstraction?
3. Does 100% test coverage imply the code being tested has no errors? Briefly explain.
4. What is the difference between a checked and an unchecked exception?
5. The method **Integer.parseInt** throws the **NumberFormatException** if the string parameter does not represent an integer value. Write a method that keeps asking the user for an integer as long as an invalid value is provided. You should catch the **NumberFormatException** and add any code needed to keep asking the user for a value.
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. Is the following code fragment legal? Briefly explain.

```
ArrayList<Boolean> L1 = new ArrayList<Boolean>();  
ArrayList<Object> L2 = L1;
```

8. 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. Add an equals method that overrides the Object class equals method.
- c. Modify the class so it implements the Comparable interface, allowing you to compare PrinterJob objects based on their id.
- d. Implement a comparator class named sizeComparator that allow us to compare PrinterJob objects based on their size.