

CMSC 132 Quiz 4 Worksheet

The next quiz for the course will be on Wed, Apr 27. The following list provides more information about the quiz:

- The quiz will be a written quiz (no computer).
- The quiz will be in lab/discussion session.
- 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. **We strongly recommend you do not use Eclipse to write the code associated with these exercises.** Try to answer the exercises in a piece of paper and then use Eclipse to verify your solutions. This approach will better prepare you for the quiz.

Exercises

1. What are two advantages to multi-threading?
2. What are two disadvantages to using multi-threading?
3. What are two ways to create threads in Java?
4. What is a daemon thread?
5. What is a data race? How can you avoid it?
6. Give an example of a Java code with a data race.
 - a. Eliminate the data race using synchronized methods, e.g., `synchronized foo( ) { ... }`
 - b. Eliminate the data race using synchronized objects, e.g., `synchronized(bar) { ... }`
7. Modify the following class so we can create threads that print messages. For the modified class, provide a main method that creates and starts two threads, one printing the message "Testudo" and the other the message "Terps". The main thread will display "UMCP" after the previous two threads have finished.

```
public class PrtMessage {
    private String message;

    public PrtMessage(String message) {
        this.message = message;
    }

    public void print() {
        for (int i=0; i<50; i++)
            System.out.println(message);
    }
}
```

8. Write Java code that computes the sum of elements in array by using two threads. One thread computes the sum of array elements with an even index; the second thread will take care of the rest. The final sum is printed by the main thread.