CMSC 131 Quiz 3 Worksheet

The next quiz for the course will be on Wed, Oct 23. The following list provides additional information about the quiz.

- The quiz will be a written quiz (no computer).
- Closed book, closed notes quiz.
- Answers must be neat and legible.
- Quiz instructions can be found at http://www.cs.umd.edu/~nelson/classes/utilities/examRules.html
- **Regarding Piazza -** Feel free to post questions in Piazza regarding the worksheet and possible solutions to problems.

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 TAs or instructor during office hours. It is recommended that you try these exercises on paper first (without using the computer).

Exercises

1. What is the difference between a static method and a non-static method?
2. When you should define a method as static?
3. What is the difference between a non-static field and a static field?
4. What does "this" represent in a class method?
5. Does a constructor for a class constructs the object?
6. Can an object live in the stack?
7. When are parameters and local variables created and destroyed?
8. What is the default value for reference and primitive type instance variables?
9. What is the default value for reference and primitive type local variables?
10. What is an immutable class?
11. Draw a memory map (as we did in lecture) that illustrates the stack and heap when the program execution reaches the point identified by /* HERE */.

```
public class Treasure {
        private String pirate;
        private int coins;

        public Treasure(String pirate, int coins) {
                this.pirate = pirate;
                this.coins = coins;
        }
        public void setCoins(int coins) {
                this.coins = coins;
        }
        public int getCoins() {
                return coins;
        }
        public String toString() {
                return "Treasure [pirate=" + pirate + ", coins=" + coins + "]";
        }
}

public class Driver {

        public static void process(Treasure c) {
                c.setCoins(2 * c.getCoins());
                /* HERE */
                System.out.println(c);
        }
        public static void search(Treasure a, int b) {
                int total = b + 10;

                b = 0;
                a.setCoins(total);
                process(a);
        }
        public static void main(String[] args) {
                int amount = 200;
                Treasure t1 = new Treasure("Pirate1", amount);

                search(t1, 3);
        }
}
```