

CMSC131, UMCP, Java Class Concepts Exercise

The goal of this exercise is to gain a better understanding of class concepts discussed in class. In a group of at least 3 students answer the questions below. To answer the questions follow this approach:

- a. One of your group members should be designed as the leader. This student will read the questions and keep moving from one question to the next. The leader will provide some reasonable amount of time for each question before moving to the next.
 - b. Before moving to the next question the group will discuss their answers.
 - c. Although you do not need to submit your answer for a grade, keep in mind that these questions can be part of midterms and quizzes.
-
1. In a Java program we have three memory areas: stack, heap and the static area. What are they? What can you find on each area?
 2. What is the difference between an instance of a class and an object?
 3. What is the difference between a static method and a non-static method?
 4. What is the difference between an instance variable and a local variable?
 5. What is the difference between an instance variable and a static variable? For example, what happens when you create objects for the Printer class below if you dropped the static from **static int count**? Draw a diagram where you create three objects assuming **static int count** and just **int count**.
 6. Why we want to define instance variables as private?
 7. How can you use private methods in order to avoid code duplication? Explain and provide an example.
 8. What is the current object? For example, assuming p1 and p2 are references to Printer objects, which object is the current object when executing p1.equals(p2)? How about when executing p1.toString()?
 9. In the equals method we accessed the printerId of the parameter using otherPrinter.printerId even though printerId is private. Why is this possible?
 10. Assume we have a public method getPrinterId(). Should we use it in the equals method in order to get the printerId of the otherPrinter object? The answer is NO, but why?
 11. What is the difference between a primitive instance variable and a reference instance variable?
 12. What is null?
 13. What would happen if we do not define any constructors? Does the class have one? Explain.
 14. Does the order of method definition matters in a class?

```

public class Printer {
    private int printerId;
    private String make;
    private static int count = 0;

    public Printer(int printerIdIn, String makeIn) {
        if (validId(printerIdIn)) {
            printerId = printerIdIn;
        } else {
            printerId = 99;
        }
        this.make = makeIn;
        count++;
    }

    public void setPrinterId(int printerIdIn) {
        if (validId(printerIdIn)) {
            printerId = printerIdIn;
        }
    }

    public String toString() {
        return "printerId: " + printerId + ", make: " + make;
    }

    public boolean equals(Printer otherPrinter) {
        if (otherPrinter == null) {
            return false;
        } else if (printerId == otherPrinter.printerId) {
            return true;
        } else {
            return false;
        }
    }

    public static int getObjCount() {
        return count;
    }

    private boolean validId(int id) {
        if (id > 0 && id < 100) {
            return true;
        } else {
            return false;
        }
    }
}

```