

# CSC 222: Object-Oriented Programming

## Test 2 Sample Questions

### True or False?

\_\_\_\_\_ One difference between Java arrays and ArrayLists is that arrays can directly store primitive values (e.g., int, char), whereas ArrayLists cannot.

\_\_\_\_\_ The number of times you can halve (and round down) a number  $N$  before it reaches 1 is roughly  $(\log_2 N)$ .

### Short Answer

A. What does it mean when we say that a class is *highly cohesive*? How does a highly cohesive design tend to lead to software that is easier to develop and reuse? Explain your answers.

B. Suppose you knew a particular algorithm was  $O(N^2)$ , and that it took 5 seconds to run on a list of 10,000 numbers. Can you predict how long that algorithm would take to run on a list of 40,000 numbers? If so, provide an estimate and describe how you obtained it. If not, explain why a prediction is not possible given only this information.

## Classes & Objects

Consider designing a program for playing a children's card game, where each card shows a number (1-8) in a color (red, black, or green). A first step in modeling a game would be to implement a class for representing individual cards:

```
public class Card {
    private int cardNum;
    private String cardColor;

    public Card(int n, String c) {
        this.cardNum = n;
        this.cardColor = c;
    }

    public int getNum() {
        return this.cardNum;
    }

    public Color getColor() {
        return this.cardColor;
    }

    public String toString() {
        return this.cardNum + "-" + this.cardColor;
    }
}
```

Trace the code segment below that creates and manipulates `Cards`. Write the output that would be produced by executing this code in the box to the right. Be specific with respect to formatting.

```
Card card1 = new Card(4, "red");

System.out.println(card1.getColor());

Card card2 = new Card(2, "black");

if (card1.getNum() > card2.getNum()) {
    System.out.println("first option");
}
else {
    System.out.println("second option");
}

Card card3 = new Card(1, "green");

ArrayList<Card> cards = new ArrayList<Card>();
cards.add(card1);
cards.add(card2);
cards.add(card3);

System.out.println(cards.size());

cards.add(0, cards.get(1));

for (int c = 0; c < cards.size(); c++) {
    System.out.println(c + ": " + cards.get(c));
}
```

