CSSE 220—Object-Oriented Software Development

Exam 1 – Part 1, Dec. 19, 2013

This exam consists of two parts. Part 1 is to be solved on these pages. If you need more space, please ask your instructor for blank paper. After you finish Part 1, please turn in your Part 1 answers and then open your computers.

Part 2 is to be solved using your computer. You will need network access to download template code and upload your solution for part 2. Please disable Lync, IM, email, and other such communication programs before beginning the exam. Any communication with anyone other than the instructor or a TA during the exam, *may result in a failing grade for the course.*

Allowed Resources on Part 1: You are allowed one 8.5 by 11 sheet of paper with notes of your choice. This section is *not* open book, open notes, and you are not allowed to use your computer for this part.

Allowed Resources on Part 2: Open book, open notes, and computer. Limited network access. You may use the network only to access your own files, the course Moodle and Piazza sites and web pages, the textbook's site, Oracle's Java website, and Logan Library's online books.

Part 1 is included in this document.

We suggest spending no more than 40 minutes on part 1.

Please, begin by writing your name on every page of the exam. We encourage you to skim the entire exam before answering any questions. Recall that *you must turn in part 1 before accessing resources for part 2. Use of your computer before turning in part 1 of the examwill be considered academic dishonesty.*

Problem	Poss. Pts.	Earned
1	9	_____
2	8	_____
3	12	_____
4	6	_____
Paper Part Subtotal	**35**	_____
Computer Part Subtotal	**65**	_____
Total	**100**	_____

Part 1—Paper Part

```java
public class Fraction {
    private int numerator;
    private int denominator;

    public Fraction() {
        this.numerator = 1;
        this.denominator = 1;
    }

    public Fraction(int num, int denom) {
        this.numerator = num;
        this.denominator = denom;
    }

    public Fraction add(Fraction frac) {
        int commonDenom = this.denominator*frac.denominator;
        int myNumerator = this.numerator*frac.denominator;
        int otherNumerator = frac.numerator*this.denominator;

        return new Fraction(myNumerator+otherNumerator, commonDenom);
    }

    public String toString(){
        return this.numerator + "/" + this.denominator;
    }
}
```

The next several questions all refer to a Fraction class. On the previous page is a listing of this class showing its fields, constructors, and methods. The javadocs are omitted to save space. DO NOT TYPE THIS CLASS IN ECLIPSE.

1. (9 points) Below are several code snippets that use the Fraction class. For each snippet, first *draw a box-and-pointer diagram* showing the result of executing it. Then *give the output* of the print statement at the end of the snippet. DO NOT TYPE THE CODE SNIPPETS FOR THIS QUESTION IN ECLIPSE.

```
Fraction first = new Fraction(2, 4);
System.out.println(first.toString());
```

(a) Output: _____
Diagram:

```
Fraction frac = new Fraction();
Fraction fracTwo = frac;
frac = fracTwo.add(new Fraction(3, 5));
System.out.println(frac.toString());
```

(b) Output: _____
Diagram:

```
Fraction[] fractions = new Fraction[3];
Fraction newFraction = new Fraction(5, 12);
for (int i = 0; i < fractions.length; i++){
    fractions[i] = newFraction;
}
fractions[1].add(fractions[1]);
System.out.println(fractions[1].toString());
```

(c) Output: _____
Diagram:

2. (8 points) Predict the output for each code snippet below. (You do *not* need to draw a diagram, but you may if it might help you.) DO NOT TYPE THE CODE SNIPPETS FOR THIS QUESTION IN ECLIPSE.

```java
int milesTraveled = 13;
int hours = 3;
System.out.println(milesTraveled / hours);
```

(a) Output: _____

```java
String[] socialMedia = new String[8];
socialMedia[0] = "Facebook";
socialMedia[1] = "Twitter";
socialMedia[2] = "Google+";
socialMedia[3] = "Instagram";
socialMedia[4] = "Linkedin";
socialMedia[5] = "Pinterest";
System.out.println(socialMedia[socialMedia.length - 5]);
```

(b) Output: _____

```java
double pi = 3.1415926535;
int num = 2;
System.out.printf( "%d QQQ %.3f", num, pi);
```

(c) Output: _____

```java
HashMap<Integer,Integer> map;
map = new HashMap<Integer,Integer>;
map.put(5,7);
map.put(11,5);
System.out.println(map.get(5));
```
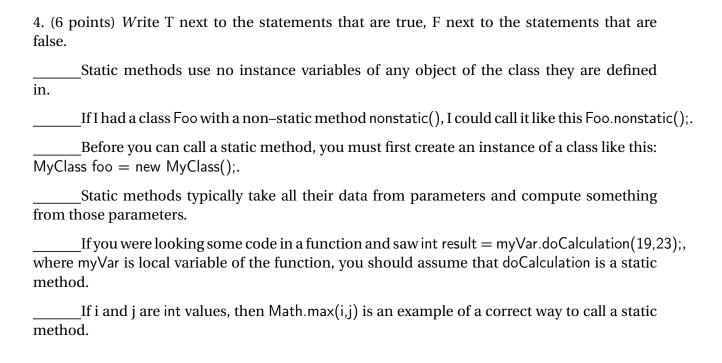
(d) Output: _____

3. (12 points) For each loop below, write down how many times its body will execute, or indicate that we can't tell from the information given. DO NOT TYPE THE CODE SNIPPETS FOR THIS QUESTION IN ECLIPSE.

```
int maxFrequency = 9;
for (int i = 0; i <= maxFrequency; i++) {
    // loop body elided
}
```

(a) Answer: _____

```
int sum = 1;
while (sum < 32) {
    sum = sum*2;
}
```

(b) Answer: _____

```
while(true){
    //generates a random number between 0 and 1
    double random = Math.random();

    if(random < 0.5) {
        break;
    }
}
```

(c) Answer: _____

```
double[] data = new double[17];
double val = 1.0;
    for (Double d : data) {
        val = d * 25;
    }
```

(d) Answer: _____

4. (6 points) *W*rite T next to the statements that are true, F next to the statements that are false.

_____Static methods use no instance variables of any object of the class they are defined in.

_____If I had a class Foo with a non–static method nonstatic(), I could call it like this Foo.nonstatic();.

_____Before you can call a static method, you must first create an instance of a class like this: MyClass foo = new MyClass();.

_____Static methods typically take all their data from parameters and compute something from those parameters.

_____If you were looking some code in a function and saw int result = myVar.doCalculation(19,23);, where myVar is local variable of the function, you should assume that doCalculation is a static method.

_____If i and j are int values, then Math.max(i,j) is an example of a correct way to call a static method.

Turn in your answers to this part of the exam before you begin the computer part.