Name: _____ Section: _____

CSSE 220—Object-Oriented Software Development

Exam 1 – Part 1, Sep. 23, 2015

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, Skype for Business, 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 for any reason, *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 or 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 45 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 exam will be considered academic dishonesty.*

Problem	Poss. Pts.	Earned
1	9	_____
2	9	_____
3	12	_____
4	5	_____
Paper Part Subtotal	**35**	_____
Computer Part Subtotal	**65**	_____
Total	**100**	_____

Part 1—Paper Part

```java
public class CalcItem {
    private double amt;

    public CalcItem(double amount) {
        this.amt = amount;
    }

    public void tripleAmount(){
        this.amt = this.amt * 3;
    }

    public double getAmount(){
        return this.amt;
    }
}

public class Calc {

    public CalcItem first;
    public CalcItem second;

    public Calc(CalcItem first, CalcItem second) {
        this.first = first;
        this.second = second;
    }

    public double calculate(){
        return this.first.getAmount() - this.second.getAmount();
    }
}
```

The next several questions all refer to the CalculatorItem and Calculator classes on the previous page 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 CalculatorItem and Calculator classes. For each snippet, first *draw a box-and-pointer diagram* (in the blank area below the snippet) showing the *final* result of executing it (i.e., you do not have to show any of the intermediate steps of any of the temporary variables created in the various methods). Then *give the output* of the print statement at the end of the snippet.

```
CalcItem a = new CalcItem(50);
Calc r = new Calc(a, new CalcItem(20));
System.out.println(r.second.getAmount() + '' '' + r.calculate());
```

(a) Output: _____

Diagram:

```
CalcItem b1 = new CalcItem(100);
CalcItem b2 = b1;
b2.tripleAmount();
Calc c = new Calc(b1, b2);
System.out.println(c.first.getAmount() + '' '' + c.calculate());
```

(b) Output: _____

Diagram:

```
CalcItem[] foo = new CalcItem[2];
CalcItem[] bar = new CalcItem[3];
bar[2] = new CalcItem(10);
foo[1] = bar[2];
bar = foo;
System.out.println(foo.length + bar.length);
```

(c) Output: _____

Diagram:

3

2. (9 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.

```
String advisor = "Advisor";
int startYear = 2014;
int startMonth = 9;
int score = (startYear - 2000) + startMonth;
System.out.println(advisor + " scores " + score);
```

(a) Output: _____

```
String[] words = {"Welcome", "alumni!", "Happy",
                  "Homecoming."};
String result = "";
for(String word : words){
    result = result + word.substring(3) + " ";
}
System.out.println(result);
```

(b) Output: _____

```
// For the given values of num and bool (true, false),
// what does invoking this method print?
public void puzzle(int num, boolean bool) {
    if (num >= 90 && bool) {
        System.out.println("Tic");
    }
    else if (num >= 20) {
        System.out.println("Tac");
    }
    else {
        System.out.println("Toe");
    }
}
```

(c) (3 points) Output:

num	bool=true	bool=false
90		
40		
20		

```
ArrayList<Integer> list = new ArrayList<Integer>();
int index = 0;
for(int i = 1; i < 11; i++){
    list.add(i);
}
while(list.size() > 0){
    // returns the element removed at location index
    Integer num = list.remove(index);
    if(num % 2 == 1){
        System.out.println(num);
    }
}
```

(d) Output: _____

4

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

```
int run = 20;
for(int i = 1; i < run; i=i+i){
    System.out.println(i);
}
```

(a) Answer: _____

```
int a = 30;
int b = 20;
while (a != b){
    if (a > b) {
        a = a - b;
    }
    else {
        b = b - a;
    }
}
System.out.println(a);
```

(b) Answer: _____

```
ArrayList<String> names = new ArrayList<String>();
names.add("Oscar");
while(names.size() != 4){
    names.add(names.get(0));
    names.add("Robinson");
}
```

(c) Answer: _____

```
int[] nums = {1, 2, 3, 4, 5};
int[] vals = {5, 4, 3, 2, 1};
int sum = 0;
for(int i = 0; i >= nums.length; i++){
    sum = sum + nums[i] + vals[i];
}
```

(d) Answer: _____

4. (5 points) *W*rite T next to the statements that are true, F next to the statements that are false.

_____Constructors in a class should be declared private.

_____Static methods cannot access instance variables or instance methods directly.

_____The keys of a HashMap are always unique, but not the values they associate with.

_____The code below prints "Hurrey!".

```
String x = ''Hurray!'';
x.replace(''a'', ''e'');
System.out.println(x);
```

_____ArrayList<int> numbers = new ArrayList<int>(); is correct code.

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