CSSE 220—Object-Oriented Software Development

Exam 1 – Part 1, Sept. 23, 2014

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 exam will 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
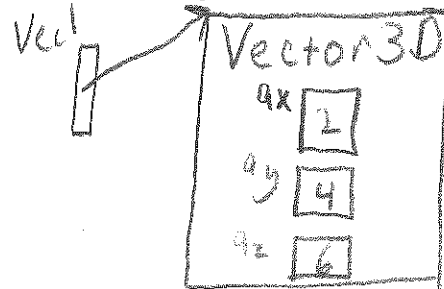
```java
public class Vector3D {
    private int ax;
    private int ay;
    private int az;

    public Vector3D() {
        this.ax = 1;
        this.ay = 1;
        this.az = 1;
    }

    public Vector3D(int xDir, int yDir, int zDir) {
        this.ax = xDir;
        this.ay = yDir;
        this.az = zDir;
    }

    public Vector3D crossProduct(Vector3D vec) {
        int cx = this.ay*vec.az - this.az*vec.ay;
        int cy = this.az*vec.ax - this.ax*vec.az;
        int cz = this.ax*vec.ay - this.ay*vec.ax;

        return new Vector3D(cx, cy, cz);
    }

    public String toString(){
        return "(" + this.ax + ", " + this.ay + ", " + this.az + ")";
    }
}
```

The next several questions all refer to a Vector3D 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 Vector3D 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.
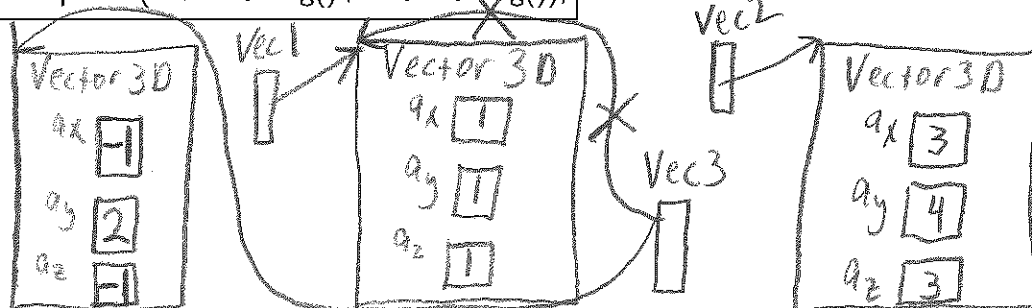
```
Vector3D vec1 = new Vector3D(2, 4, 6);
System.out.println(vec1.toString());
```
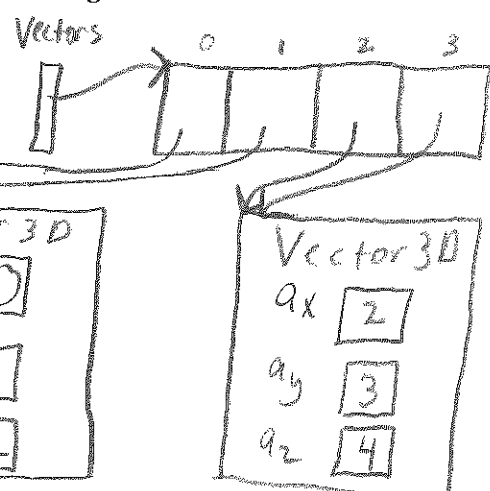
(a) Output: $(2, 4, 6)$
Diagram:

```
Vector3D vec1 = new Vector3D();
Vector3D vec2 = new Vector3D(3, 4, 3);
Vector3D vec3 = vec1;
vec3 = vec1.crossProduct(vec2.crossProduct(vec3));
System.out.println(vec1.toString()+vec3.toString());
```

(b) Output: $(1,1,1)(-1,2,-1)$
Diagram:

```
Vector3D[ ] vectors = new Vector3D[4];
for (int i = 0; i < vectors.length; i = i + 2){
    vectors[i] = new Vector3D(i, i+1, i+2);
    vectors[i+1] = vectors[i];
}
System.out.println(vectors[3].toString());
```

(c) Output: $(2, 3, 4)$
Diagram:

3

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
String name = "Johnny";
int num = 2;
int factor = 4;
System.out.println(name + num + " Shoes walked " +
                (num + factor) + " miles.");
```

(a) Output: _____

Johnny2 Shoes walked 6 miles.

```java
int[ ] nums = {1,2,3,4,5};
for(int i = 0; i < nums.length-3; i++) {
    nums[i+1] = nums[i+2] - nums[i];
}
for(int j = 0; j < nums.length; j++) {
    System.out.print(nums[j] + " ");
}
```

(b) Output: *1 2 2 4 5*

```java
Rectangle box1 = new Rectangle(1, 2, 3, 4);
Rectangle box2 = new Rectangle(1, 2, 3, 4);
if(box1 == box2)
    System.out.print("hello");
else
    System.out.print("goodbye");
```

(c) Output: *goodbye*

```java
ArrayList<Integer> nums = new ArrayList<Integer>();
while(nums.size() < 6) {
    nums.add(nums.size() - 1);
}
for (Integer current : nums) {
    System.out.print(current + " ");
}
```

(d) Output: *-1 0 1 2 3 4*

4

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.

```java
int runTimes = 18;
for (int i = 0; i <= runTimes; i++) {
    int temp = runTimes;
    temp = temp + 42;
    System.out.print(temp);
}
```

(a) Answer: **19**

```java
int a = 1;
int r = 0;
while (r < 5) {
    r = r + a;
    a = a + 1;
}
```

(b) Answer: **3**

```java
String stringOfA = "";
while(stringOfA.length() > 4) {
    stringOfA += "a";
}
```

(c) Answer: **0**
(Never enters loop.)

```java
boolean[] data = new boolean[20];
int val = 0;
while(data[val % data.length] == false) {
    data[val] = !data[val];
    val = val + 1;
}
```

(d) Answer: **20**

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

F_____Before you can call a static method in a class, you must first create an instance of that class using new.

T_____If CoolClass is the name of a class, then knowing the code "int myResult = CoolClass.coolMethod();" works proves that coolMethod must be a static method.

F_____To compare Strings, you must always use the == operator.

T_____"public CoolClass(int coolInt)" would be an appropriate declaration for a constructor in the class CoolClass.

T_____float is an example of a primative type.

T_____If i and j are int values, then the result of (i / j) will always be an int.

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