

Name: _____

USC username (e.g., ttrojan): _____

CS 455 Midterm Exam 1

Spring 2015 [Bono]

Thursday, Feb. 19, 2015

There are 5 problems on the exam, with 50 points total available. There are 7 pages to the exam, including this one; make sure you have all of them. If you need additional space to write any answers, you may use the backs of exam pages (just direct us to look there).

Note: if you give multiple answers for a problem, we will only grade the first one. Avoid this issue by labeling and circling your final answers and crossing out any other answers you changed your mind about (though it's fine if you show your work).

Put your name and USC username at the top of the exam. Please read over the whole test before beginning. Good luck!

	value	score
Problem 1	11 pts.	
Problem 2	6 pts.	
Problem 3	10 pts.	
Problem 4	8 pts.	
Problem 5	15 pts.	
TOTAL	50 pts.	

Selected methods of Java `Point` class:

`new Point(x, y)`

Constructs point object with given x and y values.

`p.translate(dx, dy)`

Changes x and y values of p by dx and dy, respectively. I.e., if p had coordinates (x, y), its new value is a point with coordinates (x+dx, y+dy)

Problem 1 [11 pts.]

Consider the following program: (Note: more about `Point` class on the cover page of the exam.)

```
public class Probl {
    public static void foo(Point a, Point b) {
        Point oldA = a;
        a = b;
        System.out.println(a + " " + b);
        b = oldA;
    }
    public static void main(String[] args) {
        Point p = new Point(5, 10);
        Point q = new Point(12, 18);
        foo(p, q);
        System.out.println(p + " " + q);
    }
}
```

Part A [7]. In the space below, draw a box-and-pointer diagram (a.k.a., memory diagram) showing all object variables, objects, and their state as they have changed during the code sequence. This includes showing `foo`'s parameters and local variable.

Part B [4]. What is printed by the code? For the purpose of this problem assume a `Point` is printed as follows: `[x, y]`

Problem 2 [6 pts.]

Consider the following static method that is supposed to return a `String` describing the weather, when given an outside temperature in Fahrenheit. (Approximately equivalent temperatures are also shown in Celsius for those of you who aren't used to Fahrenheit.) It doesn't always do the right thing.

```
public static String getWeather(int temp) {  
    if (temp >= 60) {           // 60 is about 16 C  
        return "cool";  
    }  
    if (temp >= 70) {           // 70 is about 21 C  
        return "just right";  
    }  
    if (temp >= 80) {           // 80 is about 27 C  
        return "hot";  
    }  
    if (temp >= 90) {           // 90 is about 32 C  
        return "boiling";      // FYI: "boiling" can be used to mean "very hot"  
    }  
    return "cold";  
}
```

Do not modify the code. **Show two example data values and the result of calling the method on each of them: the first one should be one where the existing method returns an incorrect weather description, and a second one such that the method returns an accurate weather description:**

<u>temp</u>	<u>return value of <code>getWeather(temp)</code></u>
-------------	--

1. (wrong)

2. (right)

Problem 3 [10 pts.]

The following method doesn't work. The method comment describes what it is supposed to do.

```
/**
 Prints the elements of the array, nums, as an addition expression followed by a
 newline. If there is only one element, prints just that number followed by
 newline. If nums has length 0, does nothing.
 Example output for various calls (shows output for arrays with 1, 2, and 3
 elements, respectively):
     5
     4 + 9
     8 + 10 + 6
 */
public static void printAdditionExpression(int[] nums) {

    for (int i = 0; i < nums.length; i++) {

        System.out.print(nums[i] + " + ");

    }

    System.out.println();

}
```

Part A [4]. Show two example data sets for which the method prints the wrong result and what the method prints in that case:

nums

output of printAdditionProblem(nums)

1.

2.

Part B [6]. Fix the code above. Do not rewrite the whole method, but rather make your changes right into the code above, using arrows to show where your code should be inserted, crossing out code that you would get rid of, etc.

Problem 4 [8 pts]

Complete the implementation of the class `TriangularGenerator`, which generates the triangular number sequence, defined below.

The k th triangular number is the sum of the numbers from 1 to k . The first six numbers in the sequence are: 1, 3, 6, 10, 15, 21. Computing T_k does not require a loop, because there is also a formula for it, shown here:

$$T_k = \sum_{i=1}^k i = 1 + 2 + \dots + k = \frac{k(k+1)}{2}$$

Furthermore, looking at the sequence, you may notice that it's even easier to generate the next triangular number if you know the previous one (no multiplications involved). **For full credit, your class must take advantage of the fact that we are generating these values in sequence, using no loops or multiplications to compute the next value in the sequence.** (You can still get partial credit for using the formula.)

The example below shows how to use the class:

```
public static void main(String[] args) {
    TriangularGenerator gen = new TriangularGenerator();
    System.out.println(gen.nextInt());    // first time returns T1 (= 1)
    System.out.println(gen.nextInt());    // 3   (1 + 2)
    System.out.println(gen.nextInt());    // 6   (1 + 2 + 3)
    System.out.println(gen.nextInt());    // 10  (1 + 2 + 3 + 4)
    . . .
}
```

[The class interface and space for your answer is given on the next page →]

Problem 4 (cont.)

The class interface and method comments appear below, you can just add in your code to complete the class:

```
//    TriangularGenerator generates the sequence of triangular numbers.  
public class TriangularGenerator {
```

```
    //    Creates triangular generator object  
    public TriangularGenerator () {
```

```
    }
```

```
    //    Returns the next triangular number in the sequence.  
    //    The first call to it returns T1 (=1).  
    public int nextInt() {
```

```
    }
```

```
}
```

Problem 5 [15 pts]

Write the static Java `int` method `largestDistance` which takes an array and returns the largest distance between two elements that have the same value. An element has a distance of zero from itself, and two adjacent elements with the same value have a distance of one. (Thus, if no two elements have the same value then you return a largest distance of 0.)

Here are some examples:

<u>arr</u>	<u>largestDistance(arr)</u>
<code>[2,3,6,10]</code>	<code>0</code>
<code>[0,0,0,0,0]</code>	<code>4</code>
<code>[2,3,2,3,2]</code>	<code>4</code>
<code>[3,2,7,7,1]</code>	<code>1</code>
<code>[4,5,100,4,5,100]</code>	<code>3</code>
<code>[1]</code>	<code>0</code>
<code>[]</code>	<code>0</code>

```
public static int largestDistance(int[] arr) {
```