Lab 20 (November 15 or November 16)

Instructions: Complete the steps below. **Be sure to show your code to one of the lab TAs before you leave, so that you can receive credit for this lab.** You must also upload copies of all your source code (.java) files to the link on Blackboard by 11:59 PM on Thursday, November 16.

1. The program defined in the "Trouble.java" file randomly crashes with one of three types of exception, or runs to completion if no exception occurs. Modify this program's `main()` method (using exception-handling) so that the program behaves as follows:

   * If calling `problem()` produces an `ArrayIndexOutOfBoundsException`, print "Alpha".

   * If calling `problem()` produces a `NullPointerException`, print "Beta".

   * If calling `problem()` produces an `ArithmeticException`, print "Gamma".

   * Otherwise, if `problem()` does not produce any type of exception, print "Delta".

   Your program should print exactly ***one*** of these four words each time it is executed (e.g., "Alpha" or "Delta", but not "Alpha" AND "Delta"). **Do not make *ANY* changes to the `problem()` method; your solution should *ONLY* change `main()` for this exercise.**

2. Define a `Triangle` class with three `private double` instance variables, each of which represents the length of one side of the triangle, and a single three-argument constructor. For any triangle, the sum of any two sides must be greater than the remaining side. Your `Triangle` class must adhere to this rule.

   Define an `IllegalTriangleException` class (an empty subclass of `Exception`), and define `Triangle`'s constructor to raise an `IllegalTriangleException` if the user attempts to create a `Triangle` whose sides violate the rule above. Your `Triangle` constructor header should resemble the following:

   ```
   /** Construct a triangle with the specified sides */
   public Triangle(double side1, double side2, double side3)
               throws IllegalTriangleException
   {
       // Implement your constructor
   }
   ```

   Finally, define a `main()` method (inside `Triangle` or in a separate driver class) that prompts the user to enter a length for each side, and then creates a new `Triangle` with those values. If the user enters an invalid set of values (meaning that the

`Triangle` constructor generates an exception), your program should use exception-handling to print an appropriate message instead of simply crashing.

Grading Guidelines: This lab is graded on a scale of 0-3 points, assigned as follows:

0 points: Student is absent or does not appear to have completed any work for the lab

1 point: Student has completed some work, but neither program compiles or runs.

2 points: Student has correctly completed only one of the programs. The second program is in-progress, but not yet functional (it may not even compile at this stage).

3 points: Student has correctly completed both programs, without any apparent errors.