1. Write a program that allows keyboard entry of three real numbers, i.e., numbers which include decimal points and which are stored as doubles in the variables a, b, c; the program should then determine if the numbers can be the lengths of the three sides of a triangle.

Note: Three positive numbers (a, b, & c) can be sides of a triangle if the sum of every two of the numbers is greater than the third.

If any of the numbers are nonpositive the program should print a single line which is formatted as follows where “<value>” should be replaced by the values provided on input.

**NOT A TRIANGLE: a = <value >    b = <value >    c = <value >     NON POSITIVE INPUTS**

If the values for a, b & c will create a triangle, the program should output a single line formatted as follows:

**TRIANGLE: a = <value >    b = <value >    c = <value >     Area of triangle = <area value>**

If a, b and c are all positive but do not make a triangle your output should be a single line as follows:

**NOT A TRIANGLE: a = <value >    b = <value >    c = <value >     : <val1> + <val2 > <= < val 3>**

Use Heron’s formula to compute the area of a triangle given the sides of the triangle.

[http://en.wikipedia.org/wiki/Heron%27s_Formula](http://en.wikipedia.org/wiki/Heron%27s_Formula)

Write two versions of the program; the first version should use the Scanner class for input and the print( ), println( ), and printf( ) methods for
output. The second version should use JOptionPane methods, i.e., GUI methods for both input and output.

Format your output to exactly three (3) decimal places after the decimal point.

2. **Before you write the program**, create five good test cases for your program. Using a calculator, determine the answers. **Do not use the instructor’s test cases.** But use the same form in preparing your test cases.

3. Compile and run your program on JGrasp. Check that your program matches your test cases and matches the test case in the text.

4. Correct any errors in your program.

5. Run instructor’s test case. Have either the lab assistant or the instructor verify that your program works on the instructor’s test case.

6. Hand in your test cases and the source code with the test case runs copied as comments onto the end of your source code.

7. **Conventions:** Follow the programming style and documentation conventions as specified in the Liang text Section 2.16 and the 110 website Project Specifications (Requirements); in case of conflict, the 110 Project Specifications will prevail. Make sure that you prompt for all user input.

8. The java source code **must** have a header formatted as follows:

    /*
    *Project #4: Triangle Calculator
    *   Section # <number>, i.e., Section 14201
    *Programmer:
    *   Date:
    *   Description:
    */