**CSE 220 Homework Assignment 1 (Due 5/30/24)**

**1) (40 pts)** Consider the following Java code snippets (i. – v.) In each case, an “expected” output is provided in red and bold. However, these code snippets will *not* print out these values – your task in each case is to explain why the provided output will not occur. *Note: in some cases, the code won’t compile/run, in others it simply won’t provide the “correct” output.* ***Be specific*** *about why the output will not occur – if you simply say “The code is wrong” will not receive much credit.  
i.* *double a = 20/16;  
 System.out.println(a);* ***Does not output “1.25”****ii.* *int import = 80;  
 int export = 40;  
 System.out.println("Net product gain is " + (import-export) + ".");* ***Does not output “Net product gain is 40.”****iii.* *int c = 10.0/2.0;  
 System.out.println(c);* ***Does not output “5” nor “5.0”****iv.* *String d = "I spent a total of " + 5 + 2 + " days waiting on the order.";  
 System.out.println(d);* ***Does not output “I spent a total of 7 days waiting on the order.”****v. String e = "The man approached us to exclaim "It is good to see you!"";*  
 *System.out.println(e);* ***Does not output “The man approached us to exclaim “It is good to see you!””***

***(See next page for Problem 2)***

**2) (30 pts)** This problem deals with the use of simple conditional and repetition statements. You will likely want to use **if** and/or **if-else** statements, will want to use a **while** (or **for)** loop, and will need to read an integer from keyboard input.

Assuming **n** is a *positive integer*, the **factorial function** **n!** is defined to return the *cumulative product of all numbers from 1 to n (i.e. )*. For example,  
4! = 1\*2\*3\*4 = 24,  
6! = 1\*2\*3\*4\*5\*6 = 720,  
etc.

You are to create class *FactorialFunc* which has a main method that does the following:  
1. Reads input integer **n** from the keyboard after a prompt.  
2a. If n is positive, calculates and prints out the factorial of n (n! as above)  
2b. If n is 0 or negative, simply prints a message to the user indicating that the code requires a positive integer to use correctly.

*Note: the two examples given for 4! and 6! above should give you a good hint as to how you can approach the calculation for part 2b above.*  
Two sample runs of the program are below:

*Input a positive number, n. I will return its factorial value (n!): 5  
5! = 120*

*Input a positive number, n. I will return its factorial value (n!): -4  
Sorry, but the value for n must be positive.*

***(See next page for Problem 3)***

**3) (30 pts)** *This problem deals with reading input from i. a keyboard and ii. a file, making simple calculations, and returning the results.*

In geometry, we define the **semiperimeter** of a polygon to be *half of its perimeter.* This measure is of particular importance when discussing triangles, because it provides an easy means of computing a triangle’s area using **Heron's formula** according to the three steps below:  
1. Let **a**, **b**, and **c** represent the three sides of any triangle.  
2. Semiperimeter **s** of the triangle can then be computed as   
3. The area of the triangle is then computed as

i. Create class *TriangleKBInput* which has a main method that does the following:  
– Reads the 3 sides (**a**, **b**, **c**) of a triangle from keyboard after prompting the user.  
– *Computes* and *displays* the triangle’s semiperimeter **s**.  
– *Computes* and *displays* the **area** of the triangle.  
**Note: your program should accommodate decimal input and output values.**

A sample run of the program should look something like the below:  
*Enter triangle side a: 6.5  
Enter triangle side b: 7.5  
Enter triangle side c: 10  
The triangle has semiperimeter of 12.0  
The triangle has area 24.3721152139.*

ii. Create class *TriangleFileInput* which has a main method that acts synonymously to TriangleKBInput, but reads the three sides of the triangle from a file called “TriangleSides.txt” (which is included with this assignment posting) and displays the semiperimeter/area values.   
A sample run of the program should look something like the following:  
*Reading input from file...  
The triangle has a semiperimeter of 6.25.  
The triangle has an area of 6.051536478449089.*

**Responses to Problem #1 should be in .doc(x) or .pdf format. Upload a zip file containing this file and your .java files for problems 2 and 3 to Blackboard. The zip file (not the .java files) should have the filename “LN\_FN\_1.zip” where LN is your last name and FN is your first name.**