University of Victoria
Department of Computer Science
CSC 110 Fundamentals of Programming
ASSIGNMENT 1

DUE: January 19, 2018 before 11:50 pm – By submission on ConneX only

Assignment Instructions

How to hand in your work

Submit the requested files for parts (b) and (c) (see below) through the Assignment #1 link on the CSC 110 conneX site. Please make sure you follow all the required steps for submission (including the final confirmation of your submission). Part (a) is not marked.

Marking

Your mark will be based on:

- Your code compiling and running.
- Your code producing the output exactly as requested, and calculating the object's dimensions correctly.
- Your code indented and appropriately documented via comments. (Please follow the guidelines in codingConventions.pdf available in the "Notes" section in the "Resources" on conneX.)

Part (a): Problems from the Textbook

Complete the Chapter 1 Self--Check Problems 1 to 20 and compare your answers to those given by the textbook authors at: http://www.buildingjavaprograms.com/self-check-solutions-4ed.html

Part (b): ASCII Art

Write a program that creates a version of the Canadian flag using ASCII art, as follows:

```
|--------------------------------------|
|**********                  **********|
|**********         ^        **********|
|**********       ^ /*\ ^     **********|
|**********      /*\|*|/*\    **********|
|********** . --**********-- . **********|
|**********    \*********/    **********|
|**********     >*******<     **********|
|**********    ***********    **********|
|**********     ---------     **********|
|**********         ||        **********|
|**********                   **********|
|--------------------------------------|
```

This looks best when your terminal uses a monospaced font such as Courier.
It is composed of the following characters: space, asterisk (*), forward slash (/), back slash (\), carot (^), vertical line ('|") and the hyphen('-').

Write a Java file that creates this ASCII art flag as output.

File to submit: **`Flag.java`**

Part (c): Math calculations

The number Pi (π) is the ratio of any circle's circumference to its diameter. Researchers are constantly searching for a more accurate representation of π. In many programming libraries, however, this number is approximated with the following calculation:

$\pi = 4 \times (1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + 1/13 - 1/15 + \ldots)$

(Recall from math that . . . means that this series is infinite or goes on forever.)

1. (Hand Calculations – Simplified Test Case) Use a calculator and paper/pencil to determine an approximation for π using only the first 8 terms of the sequence above.

   $\pi = 4 \times (1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + 1/13 - 1/15) =$ _____

   Write down as many decimal points as your calculator provides for you, then re-check your calculation several times.

2. (Program – Simplified Program) Write a Java program that approximates π using the first 8 terms of the sequence.

   Although this can be done by simply typing the numbers 1 - 1/3 + 1/5 – 1/7 + 1/9 – 1/11 + 1/13 – 1/15 into your editor then multiplying by 4 (and you can try it that way at first), do not move on to step 3 (below) until you have written your program using the following to create this series:

   a. Create a variable called nextTerm and assign it (initially) the value 1.
   b. Create a variable called denominator and assign it (initially) the value 1.
   c. Create a variable called series and assign it (initially) the value 0.
   d. Do the following 8 times:
      i. Add (nextTerm\denominator) to the series.
      ii. Add 2 to the denominator.
      iii. Multiply the nextTerm by -1.

   Multiply the series by 4 then output the result.

   Sample output of this program:

   ```
   Pi is approximately 3.017071817071818
   ```

3. (Complete Program) Write a Java program that approximates π using the first 400 terms of the sequence above.

   Sample output of this program:

   ```
   Pi is approximately 3.1390926574960143
   ```

File to submit: **`ApproxPI.java`**

Grading scheme

- → A grade: An exceptional submission demonstrating creativity and initiative. The two programs run without any problems, have the expected output and follow the coding conventions beautifully.
- → B grade: A submission completing the requirements of the assignment. The two programs run without any problems and have the expected output.
- → C grade: A submission completing most of the requirements of the assignment. The programs run with some problems OR only one program runs without any problems and has the expected output.
- → D grade: A serious attempt at completing requirements for the assignment. The two programs run with quite a few problems OR only one program runs with some problems.
- → F grade: Either no submission given, the submission does not compile, or submission represents very little work.