CSE1341 – Lab #2

PRE-LAB [10 points]
Must be done prior to your lab session .

Create a Java program named Triangle.java that prompts for the base and the height of a triangle. It then calculates and displays the area of the triangle.

```
$ java Triangle
Please enter a value for the base of a triangle? 4
Please enter a value for the height of the triangle? 5
The area of the triangle is 10.0
```

Bring the compiled program (*.class file)* along with the *.java* source file to your lab session. Demonstrate the working program to your lab instructor for pre-lab credit.

LAB [90 points]

NOTES: Use the given notes as a guide for the program logic. These comments must
be included in the programs to explain the logic followed.
Each program should compile without errors and should run to produce outputs
described for each exercise. The following points will be discounted if the related
element is missing or incorrect:

Reasonable output formatting [20 points]
Proper names for classes and variables [15 points]
Comments [15 points]
Program doesn't compile [20 points]
Source code (java file) missing [15 points]
Executable (class file) missing [15 points]

The following programs are all based on the spreadsheet assignments you completed for lab 1.
Complete the code for each of them so they compile and run successfully.

1) [25 points] The Fibonacci sequence is a pattern of integers starting with zero and one, where each subsequent value is equal to the sum of the prior two values:

$$F_n = F_{n-1} + F_{n-2}, \text{ where } F_0 = 0 \text{ and } F_1 = 1.$$

Create a Java program which will prompt the user with "How many numbers do you want to have in the sequence? (It must be greater than 2.)". Then it will prompt the user with "Do you want to skip the odd numbers? 1=yes, 2=no".

The program will then run according to the inputs from the user. That is, if the user enters '12' and '2', then the program will just print the first 12 numbers in the sequence. If the user enters '12' and '1', then the program will try to print the first 12 numbers but will replace the odd numbers with blanks as shown below.

Sample runs:

```
$java Fibonacci
Fibonacci Sequence:
How many numbers do you want to have in the sequence?
(It must be greater than 2.)
12
Do you want to skip the odd numbers? 1=yes,2=no
2
     1     0
     2     1
     3     1
     4     2          Sample
     5     3
     6     5
     7     8
     8     13
     9     21
    10     34
    11     55
    12     89
```

```
$java Fibonacci
Fibonacci Sequence:
How many numbers do you want to have in the sequence?
(It must be greater than 2.)
12
Do you want to skip the odd numbers? 1=yes,2=no
1
     1     0
     2
     3                Sample
     4     2
     5
     6
     7     8
     8
     9
    10     34
    11
    12
```

2) [20 points] World Population

The total world population was estimated at 7.6440 billion people in August 2018. Assuming an annual growth rate of 1.09% per year, create a Java program showing the world population in 2018 and for each of the next 25 years. Verify your results using the spreadsheet you created for Lab #1.

Hint: if calculated properly, the U.S. population in 2043 would be 10.0236 billion people.

```
$ java Population
Year    World Population (billions)
------------------------------------
2018            7.6440000
2019            7.7273196
2020            7.8115474
2021            7.8966933
2022            7.9827672
2023            8.0697794
2024            8.1577400
2025            8.2466593
2026            8.3365479
2027            8.4274163
2028            8.5192751
2029            8.6121352
2030            8.7060075
2031            8.8009030
2032            8.8968328
2033            8.9938083
2034            9.0918408
2035            9.1909419
2036            9.2911231
2037            9.3923964
2038            9.4947735
2039            9.5982665
2040            9.7028876
2041            9.8086491
2042            9.9155634
2043           10.0236430
```

3) [25 points] Create a Java program named UnitConversion.java that converts the following:

1. Liter to gallon
2. Gallon to liter
3. Kilometers to miles
4. Miles to kilometers

The conversion formulas:

1 liter	=	0.264172 gallon
1 gallon	=	3.78541 liters
1 kilometer	=	0.621 mile
I mile	=	1.609 kilometers

Use the following sample data in your program - for grading purpose.

```
$ java UnitConversion
1. liters to gallons:
Please enter a value for liters:40
40.0 liters  =  10.57 gallons

2. gallons to liters:
Please enter a value for gallons:40
40.0 gallons =  151.42 liters
                    Sample
3. km to miles:
Please enter a value for kilometers:400
400.0 km  =  248.40 miles

4. miles to km:
Please enter a value for miles:400
400.0 miles  =  643.60 km
```

4) [20 points] Compound Interest Calculator

The formula for annual compound interest, including principal sum, is:

$$A = P \left(1 + r/n\right)^{(nt)}$$

Where:

A = the future value of the investment/loan, including interest

P = the principal investment amount (the initial deposit or loan amount)

r = the annual interest rate (decimal)

n = the number of times that interest is compounded per year

t = the number of years the money is invested or borrowed for

Create a Java program named CompundInterest.java which prompts the use to enter P, r, n, t and find the future value of the investment/loan, including interest (A) and the total interest earned. Use the data in Lab 1 to test your program.

(Lab1: Assume you put $120,000 into a bank. How much will your investment be worth after 15 years at an annual interest rate of 4% compounded quarterly? How much will be the total interest earned?)

Use the following sample format to show your output - for grading purpose.

```
$ java CompoundInterest
Welcome to the compound interest calculator!

Please enter the principal: 120000
Please enter the annual interest rate: 4
Please enter the compounding periods per year: 4
Please enter the term in years: 15
                                      Sample
Amount Earned (A) : $218,003.6
Total Interest Earned: $98,003.6
```

Submit the *java* and *class* files via Canvas (as a single zip-file). Include a comment block at the top of each *Java* file that includes your name, student id number, and "Lab 2-Fall 2018".

This assignment is due by 6:00AM Saturday, September 22.