CS 5000 – Spring 2020
Assignment #3

Math Methods, Characters, and Strings – Chapter 4

Develop a complete Java program for each of the following problems. Please name the programs as indicated and add proper program headers and output labels as specified. ***Do not use loops, arrays, methods, or other concepts we did not discuss to date***, ***please build on what we have covered to date.***

Make sure you **include a header for each program (see previous assignments).**

Program #1 (10 points): Write a Java program (name it `StringMethods`) that reads from the user two strings (say `string_1`, and `string_2`). The program uses String class methods to manipulate these input strings as follows and prints out the outcome as shown below.

a) Determine the length of string_1.
b) Determine the length of string_2.
c) Concatenate both strings, separated by space.
d) Check if the two strings have same set of characters with regard to case (i.e., equal).
e) Convert string_1 to upper case.
f) Convert string_2 to lower case.
g) Extract a valid sub-string of multiple characters from string_1.

Make sure to properly label your output prompts for each manipulation above and print the outputs on separate lines. Use the escape character (\t) to line-up the outputs after the labels as follows (for inputs: `John` and `Amy`):

```
a) Length of String 1:    4 characters
b) Length of String 2:    3 characters
c) Concatenation:         John Amy
d) Equal Strings?         Not equal
e) Uppercase String 1:    JOHN
f) Lowercase Sting 2:     amy
g) Valid substring:       oh
```

Separate your code into sections with proper in-line comment before each section, such as

```
// Part A: Determine the length of string_1
```

Program #2 (10 points): Write a Java program (name it `RandomNumbers`) that generates random numbers as follows.

a) A random integer number between 20 and 80 (inclusive).
b) A random integer number between -20 and 20 (inclusive).
c) A random integer number between -50 and -20 (inclusive).
d) A random floating-point number between 0.0 and 21.9999 (inclusive).

Properly label output for each part above, print the outputs on separate lines, and use the tab escape character (\t) to line-up the outputs after the labels as shown in this sample run:

```
a) A random integer between 20 and 80 (inclusive):       51
b) A random integer between -20 and 20 (inclusive):      -9
c) A random integer between -50 and -20 (inclusive):     -31
d) A random float between 0.0 and 21.9999 (inclusive):   19.9048
```

Separate your code into sections with proper in-line comments such as

```
// Part A: Generate random integer number between 15 and 65 (inclusive)
```

Program #3 (10 points): Write a java program (name it `Vowel.java`) that reads from the user a letter (**char type, not String type**), then determines if the letter is either a vowel, or a consonant, or invalid input (neither vowel nor consonant). Notice that the user can enter any character on the keyboard. Document your code and format the input and output labels following these example runs:

First run:

```
Please enter a letter: D
You entered D
D is a consonant
```

Second run:

```
Please enter a letter: e
You entered e
e is a vowel
```

Third run:

```
Please enter a letter: ;
You entered ;
; is invalid input
```

Program #4 (10 points): Write a Java program (name it `Payroll`) that reads the following information about an employee and prints out a payroll statement as shown in the sample run below. Run and test the program with different input values and check your math manually to validate the outputs. Document the code and format input prompt and output labels as shown below. Use escape characters and formatting object (for $) as needed to lineup the outputs as shown below.

```
Enter employee's name: John Smith
Enter weekly hours worked: 40
Enter hourly pay rate: 10.0
Enter federal tax rate: 0.1
Enter state tax rate: 0.05

Payroll statement
=================
Employee Name:          John Smith
Hours Worked:           40
Pay Rate (per Hour):    $10.00
Gross Weekly Pay:       $400.00
Federal Withholding:    $40.00
State Withholding:      $20.00
Net Pay:                $340.00
```

Program #5 (10 points): Write a Java program (name it `MathMethods`) that uses the **`MATH`** class methods to perform the following tasks.

a) Prompt the user to enter a negative integer number and print out its absolute value.
b) Prompt the user to enter a floating-point number representing an angle and print out the angle's cosine, sine, and tangent values.
c) Prompt the user to enter a floating point number and print out both of its floor and ceiling values.
d) Prompt the user to enter two floating-point numbers (say X and Y) and print out the value of X^Y (X is raised to the power of Y).
e) Prompt the user to enter an integer number and print out its square root. Try that with both negative and positive inputs.

Make sure to properly label your output for each task and print the outputs on separate lines. Use the tab escape character to line-up the outputs after the labels. See sample run below.

```
Enter a negative integer number: -5
The absolute value of -5 is: 5

Enter a floating-point number representing an angle: 0.45
The cosine of this angle is:   0.90044710753787
The sine of this angle is:     0.43496552337706423
The tangent of this angle is:  0.48305505091399376

Enter a floating point number: 5.39
The floor value of 5.39 is:    5.0
The ceiling value of 5.39 is:  6.0

Enter a floating-point number as X: 2.0
Enter a floating-point number as Y: 3.0
The value of X^Y is: 8.0

Enter an integer number: 16
The square root of 16 is: 4.0
```

Separate your code into sections with proper in-line comments such as

```
// Part A: Enter a negative number and print its absolute value
```

Submission:

1. Before submitting your programs, make sure you review the assignment submission requirements and grading guidelines on the course webpage. The grading guidelines explain some of the common errors found in programming assignments.
2. The assignment is due no later than **5:00pm** on the due day posted in D2L.
3. Please compile and run your java files (only the .java files) right before you upload to the assignment submission folder in D2L.