Concordia University
Computer Science & Software Engineering
Faculty of Engineering and Computer Science

Concordia University
COMP 248 – Fall 2016
Assignment 3

Due Date:	By 11:55pm November 9th , 2016
Evaluation:	4% of final mark (see marking rubric at the end of handout)
Late Submission:	none accepted
Purpose:	The purpose of this assignment is to help you learn Java flow of control statements (nested loops), arrays of primitive types, arrays of strings, 2-dimensional arrays.
CEAB/CIPS Attributes:	Design/Problem analysis/Communication Skills

General Guidelines When Writing Programs:
Please refer to the handout of Assignment #1.

Question 1(One dimensional array)
You are hired to invent a **Mad Libs** like game; but this version is for some amusing *POEM WRITING*. You are implementing a simple prototype to write a *POEM* that is inspired by *wilderness* and the *nature* theme. Your program will ask the user to enter a set of relevant nouns and adjectives that will be used to generate different versions of a poem. You can assume each poem has 3 lines.
Your program should proceed as follows:

1. Display a welcome message and ask the user to enter a set of nouns and adjectives. Store the nouns in one array and the adjectives in another array for further processing. The user should enter 3 nouns and 3 adjectives at least.

```
         -------------------------------------------------
          Do you want to write a poem?? Lets get started!!
         --------------------------------------------------

How may nouns ? min 3
8
Enter 8 nouns
mountain
Fountain
sea
Tree
BEE
showers
Flowers
birds
How may adjectives? min 3
6
Enter 6 adjectives
beautiful
Vast
majestic
refreshing
impeccable
breathtaking
```

2. Generate your poem by choosing at random combinations of noun and adjectives.
3. You are not allowed to pick a noun or an adjective again until you have used all sets of noun and adjectives that the user has provided, respectively.
4. Your poem should be displayed in the format below (regardless how the user enters the nouns/adjectives)
5. The user can proceed by choosing different versions of the poem as follows:

```
        ------------------------------
            Here is my Java POEM!!
            ****LOOK AROUND****
        ------------------------------

        BREATHTAKING SHOWERS
                REFRESHING TREE
                        IMPECCABLE SEA


Another Poem (y/n)? y

        ------------------------------
            Here is my Java POEM!!
            ****LOOK AROUND****
        ------------------------------

        MAJESTIC MOUNTAIN
                BEAUTIFUL BEE
                        VAST FLOWERS


Another Poem (y/n)? y

        ------------------------------
            Here is my Java POEM!!
            ****LOOK AROUND****
        ------------------------------

        BEAUTIFUL FOUNTAIN
                IMPECCABLE BIRDS
                        BREATHTAKING MOUNTAIN


Another Poem (y/n)? y

        ------------------------------
            Here is my Java POEM!!
            ****LOOK AROUND****
        ------------------------------

        REFRESHING BIRDS
                MAJESTIC FLOWERS
                        VAST TREE


Another Poem (y/n)?
```

6. When the user chooses to stop, your program should display a closing message like: *‘ Thank you for using our JAVA POEM GENERATOR’*

Another possible output of the program is shown as follows:

```
-------------------------------------------------
         Do you want to write a poem?? Lets get started!!
        --------------------------------------------------

How may nouns ? min 3
2
How may nouns ? min 3
3
Enter 3 nouns
roses
grass
tulipes
How may adjectives? min 3
1
How may adjectives? min 3
4
Enter 4 adjectives
red
lovely
Gorgeous
green

        ------------------------------
            Here is my Java POEM!!
            ****LOOK AROUND****
        ------------------------------

        GREEN GRASS
                RED TULIPES
                        LOVELY ROSES


Another Poem (y/n)? n
```

Note: user input is highlighted in grey.
Hint: You will need to use the random() function – as in Assignments 1&2- to choose your nouns and adjectives. Keep track of your choices so that you do not pick the same noun or the same adjectives again, unless the entire list of nouns and adjectives has been used.

<u>Question 2</u> – 2 dimensional arrays

You are requested to write a program to book tickets for a play. Assume a small theatre having 6 rows each with 7 seats/row) labeled A, B, C, D, E, F and G.

The seats prices are as follows:

- 'Gold tickets' = seats in rows 1 and 2 cost 100 $/seat
- 'Silver tickets' = seats in rows 3 and 4 are cost 70 $/seat
- 'Bronze tickets' = seats in rows 5 and 6 are cost 40$/seat

Your program should proceed as follows:

1. Display a welcome message, available seats and prices

```
        ------------------------------------------------
        --------COMP248 Concert IS BACK IN TOWN------------
                 Hurry book your ticket Now!!
        --------------------------------------------------

1 A B C D E F G
2 A B C D E F G
3 A B C D E F G
4 A B C D E F G
5 A B C D E F G
6 A B C D E F G

Rows 1 & 2 Gold   100 CAD/ticket
Rows 3 & 4 Silver  70 CAD/ticket
Rows 5 & 6 Bronze  40 CAD/ticket
Available seats 42
How many tickets do you need?
```

2. The user is allowed to book several tickets in one session. The program needs to check if there are enough empty seats to proceed with the booking.
3. The program needs to verify that the user is entering a valid row number and a valid seat letter.
4. For every seat request, the program displays the balance to be paid and marks the reserved seat with 'X' if the seat is available (otherwise the user needs to choose another seat).
5. When the user is successful with booking his seats, the system prompts him with a message that his reservation is complete.
6. Now a second user can start a new booking session (repeat steps 2-5).
7. If no more seats are available the program displays a corresponding message and terminates

Here is an example of a booking session:

```
How many tickets do you need?
2
Input your seat selection
Enter the row number and then the seat letter (example: 3B)
3C
Your seat is reserved. Your balance is 70.0
1 A B C D E F G
2 A B C D E F G
3 A B X D E F G
4 A B C D E F G
5 A B C D E F G
6 A B C D E F G
Input your seat selection
Enter the row number and then the seat letter (example: 3B)
3D
Your seat is reserved. Your balance is 140.0
1 A B C D E F G
2 A B C D E F G
3 A B X X E F G
4 A B C D E F G
5 A B C D E F G
6 A B C D E F G
Reservation complete! Please proceed to payment
```

As follows is an example of a new booking session:

```
Do you wish to start a new booking? (y/n)?
y
1 A B C D E F G
2 A B C D E F G
3 A B X X E F G
4 A B C D E F G
5 A B C D E F G
6 A B C D E F G

Rows 1 & 2 Gold   100 $/ticket
Rows 3 & 4 Silver  70 $/ticket
Rows 5 & 6 Bronze  40 $/ticket
Available seats 40
How many tickets do you need?
1
Input your seat selection
Enter the row number and then the seat letter (example: 3B)
1G
Your seat is reserved. Your balance is 100.0
1 A B C D E F X
2 A B C D E F G
3 A B X X E F G
4 A B C D E F G
5 A B C D E F G
6 A B C D E F G
Reservation complete! Please proceed to payment
Do you wish to start a new booking? (y/n)?
```

Here are examples of some error messages that the program should display

```
Do you wish to start a new booking? (y/n)?
y
1 A B C D E F X
2 A B C D E F G
3 A B X X E F G
4 A B C D E F G
5 A B C D E F G
6 A B C D E F G

................
Available seats 39
How many tickets do you need?
3
Input your seat selection
Enter the row number and then the seat letter (example: 3B)
3C
Sorry seat is not available
Input your seat selection
Enter the row number and then the seat letter (example: 3B)
7G
Invalid row number -try again
Input your seat selection
Enter the row number and then the seat letter (example: 3B)
EE
Invalid row number -try again
Input your seat selection
Enter the row number and then the seat letter (example: 3B)
1P
Invalid Seat letter - try again
Input your seat selection
Enter the row number and then the seat letter (example: 3B)
Anything
Invalid Seat assignment
Input your seat selection
Enter the row number and then the seat letter (example: 3B)
......................
```

You should not proceed with the booking if the requested seats are more than the availability.

```
Do you wish to start a new booking? (y/n)?
y
1 X X X D E F X
2 X X X X X X X
3 A B X X E F G
4 X X X X X X X
5 X X X D E F G
6 X X X X X X X
...................
Available seats 12
How many tickets do you need?
15
Sorry cannot process your request
No more seat are available
Thank you for choosing our reservation system!!
```

Note:
- Use a 2D array for seat assignments.
- You can use static methods if you like.
- User input is highlighted in grey.

Question 3 – Defining a class

a) Create a class named `Scale` that stores information about the body weight. It contain the following:

- Private instance variables that store the value of the weight reading and the unit used (kg or lbs).

- 3 constructors:
  - No-argument constructor (sets weight to zero and unit to kg)
  - One argument constructor (sets the weight to a value and assumes unit to be kg)
  - Two argument constructor (sets both weight and unit)

- 2 accessor methods:
  - method to return the value of the weight in kg
  - method to return the value of the weight in lbs

  (Note 1kg=2.2 pounds , 1 pound= 1/2.2 kg)

  Assume both methods will return values of only 2 decimal points.
- 2 mutator methods:
  - method to set the value of the weight
  - method to set both weight and unit

- A public method called `waterIntake()` that returns an integer value representing the number of recommended daily cups of water intake given the weight.

  A rule-of-thumb to calculate the recommended daily water intake, is to take half your body weight(in pounds), and drink that amount in ounces of water. Assume a cup is around 8 ounces.
  It is recommended to add 12 ounces of water to your daily intake for every 30 minutes of work out. Write another version of the method `waterIntake()` that takes an argument for the daily minutes of exercise and returns the daily recommended water intake (in cups).

- A `toString()` method to return the value and the unit of the weight
- An equals() method to test for equality of two objects of class `Scale`.
- `isLessThan() and isGreaterThan() method to compare between` two objects of class `Scale`.

b) Write a test code

I. Which declares 3 weights objects using 3 different constructors and output the description of the 3 weights (include 1 weight in kg and 1 weight in pounds)

II. Test your accessor methods to convert weights to lbs or kg

III. Calculate the recommended water intake, if daily exercise hours are given or not (include 1 weight in kg and 1 weight in pounds)

IV. Use the mutator methods to change weight -value and unit- and test your methods `equals()` , `isLessThan()` and `isGreaterThan()` for different cases of Scale objects as shown below.

```
Weight 1: 0.0 kg
Weight 2: 50.0 kg
Weight 3: 110.0 lbs
```

i. Create weight 1,2 and 3

```
Weight 2 in pounds: 110.0
Weight 3  in kg: 50.0
```

ii.Convert weight 2 to pounds
Convert weight 3 to kg

```
Daily water intake for  50.0 kg = 6 cups
Daily water intake for  110.0 lbs = 6 cups
Daily water intake for  50.0 kg and 90min workout= 11 cups
Daily water intake for  110.0 lbs and 60min workout= 9 cups
```

iii. Calculate water intake for weight 2 &3

```
Weight 1 & Weight 2 0.0 kg & 50.0 kg =? false
Weight 2 & Weight 3 50.0 kg & 110.0 lbs =? true
```

Test if weights 1,2 & 3 are equal

```
Modify weight 1: 50.0 kg
Weight 1 50.0 kg Weight 2 50.0 kg =? true
Weight 1 50.0 kg &  Weight 3 110.0 lbs =? true
```

Modify weight 1 and compare with weights 2 & 3

```
Weight 4: 50.0 lbs
Weight 4 & Weight 2 =? 50.0 lbs =? 50.0 kg false
Weight 4 & Weight 3 =? 50.0 lbs =? 110.0 lbs false
```

Create weight 4 and compare to weights 2 & 3

```
Weight 2 < Weight 4 ? 50.0 kg < 50.0 lbs  false
Weight 3 < Weight 4 ?110.0 lbs < 50.0 lbs false

Weight 2 > Weight 4 ? 50.0 kg > 50.0 lbs true
Weight 3 > Weight 4 ? 110.0 lbs > 50.0 lbs true

Weight 1 < Weight 2 ? 50.0 kg < 50.0 kg false
Weight 1 > Weight 3 ? 50.0 kg > 110.0 lbs false
```

Compare weights using your `isLessThan()` and `isGreaterThan()` methods

Submitting Assignment 3

Please check your course moodle webpage on how to submit the assignment.

Evaluation Criteria for Assignment 3 (20 points)

Source Code		
Programming Style (3 pts.)		
Comments-description of variables and constants-description of the program (authors, date, purpose)	0.5	pt.
Clear prompts to user & clear message with output	1	pt.
Use of significant names for identifiers	0.5	pt.
Indentation and readability	0.5	pt.
Welcome Banner/Closing message	0.5	pt.
Question 1 (5 pts.)		
Reading data and verifying input	1	pt.
Selection for Poem lines/variation in nouns-adjectives	2	pt.
Generate different versions of poem	1	pt.
Display poem properly formatted	1	pt.
Question 2 (7 pts.)		
Initialization and display of booking layout	1	pt.
Reading seat assignment and verifying data	2	pt.
Checking seat availability	1	pt.
Make reservation and display price	1	pt.
Update and display booking layout	1	pt.
Repeat for different booking sessions	1	pt.
Question 3 (5 pts.)		
Constructor-accessor-mutator	1.5	pt.
Comparison (=,<,>) , string	1	pt.
Other methods	0.5	pt.
Test code	2	pt.
TOTAL	**20**	**pts.**