IS 380: Object-Oriented Programming
Spring 2017

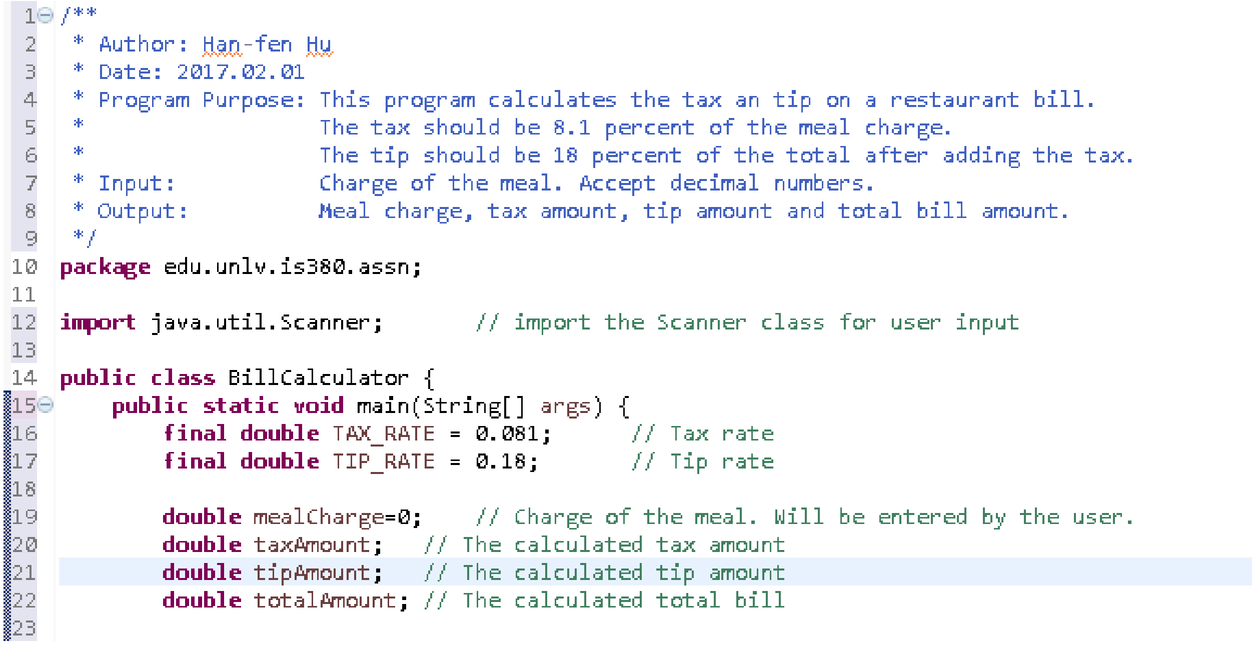
Individual Assignment 2

Due Date: **3:59 PM, February 22, 2017** (Submit via WebCampus).
Weights: 5% of total grades.

NOTE:

1. Please zip the **.java** files and upload the zip file to WebCampus for submission.
2. If you use Eclipse to complete the assignment, go to the workspace you specified when you launched Eclipse (For example, D:\workspace), and find the project folder you created and then the \src folder. Go down the file directories (i.e., the package structure) until you find the .java files there.
3. Please provide proper comments to document your code, including the following:
   a. Author’s name;
   b. Purpose of the program;
   c. Input data;
   d. Output data;
   e. In-line comments for the statements, including the variable/constant declaration and initialization.

****This is an example of the program comments. This is NOT a complete program.**



```java
/**
 * Author: Han-fen Hu
 * Date: 2017.02.01
 * Program Purpose: This program calculates the tax an tip on a restaurant bill.
 *                  The tax should be 8.1 percent of the meal charge.
 *                  The tip should be 18 percent of the total after adding the tax.
 * Input:           Charge of the meal. Accept decimal numbers.
 * Output:          Meal charge, tax amount, tip amount and total bill amount.
 */
package edu.unlv.is380.assn;

import java.util.Scanner;       // import the Scanner class for user input

public class BillCalculator {
    public static void main(String[] args) {
        final double TAX_RATE = 0.081;      // Tax rate
        final double TIP_RATE = 0.18;       // Tip rate

        double mealCharge=0;    // Charge of the meal. Will be entered by the user.
        double taxAmount;   // The calculated tax amount
        double tipAmount;   // The calculated tip amount
        double totalAmount; // The calculated total bill

```

Questions

1. (30 points) Write a program that computes the income tax for an individual. The program should ask the user to enter the total taxable income of the year. The program then uses the tax bracket (as shown below) to calculate the tax amount:

    10% on taxable income from $0 to $9,325, plus
    15% on taxable income over $9,325 to $37,950, plus
    25% on taxable income over $37,950 to $91,900, plus
    28% on taxable income over $91,900 to $191,650, plus
    33% on taxable income over $191,650 to $416,700, plus
    35% on taxable income over $416,700 to $418,400, plus
    39.60% on taxable income over $418,400

    (1) The program should display the total tax due to the user.
    (2) The program should show an error message if the user enters a negative number.

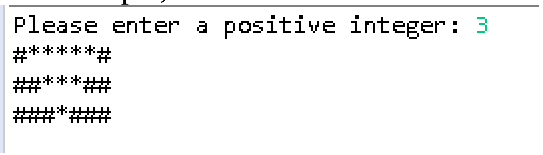
Grading criteria:

(1) Correctness (20 points):
- (a) The code can be compiled without any syntax error.
- (b) The code can generate the requested results. Please make up some numbers to test the program and validate the results.
- (c) The program is documented using comments.

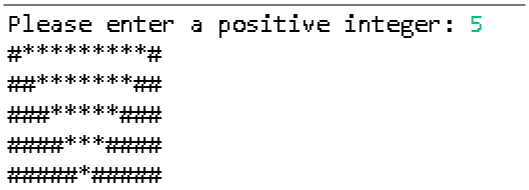
(2) Technique used (10 points):
- (a) Variable and constant declaration and initialization is done following the naming convention of Java programs. (Note: Constant must be used in this program).
- (b) Getting user input is done. (You can use the text at the console or the dialog boxes)
- (c) Arithmetic operators are used.
- (d) Showing output is done. (You can use the text at the console or the dialog boxes)
- (e) **`if-else`** or **`switch`** statements should be used to complete the program.

2. (30 points) write a program that draws a certain pattern. Based on the number entered by the user, display a set of (reversed) triangles on the screen. If a negative number is entered, show an error message.

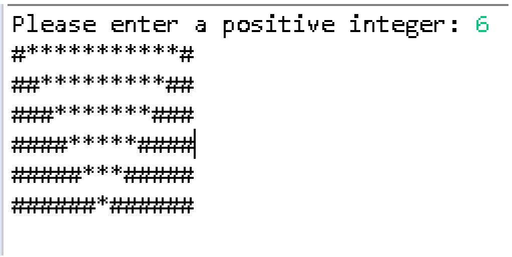
    For example,



```
Please enter a positive integer: 3
#*****#
##***##
###*###
```



```
Please enter a positive integer: 5
#*********#
##*******##
###*****###
####***####
#####*#####
```

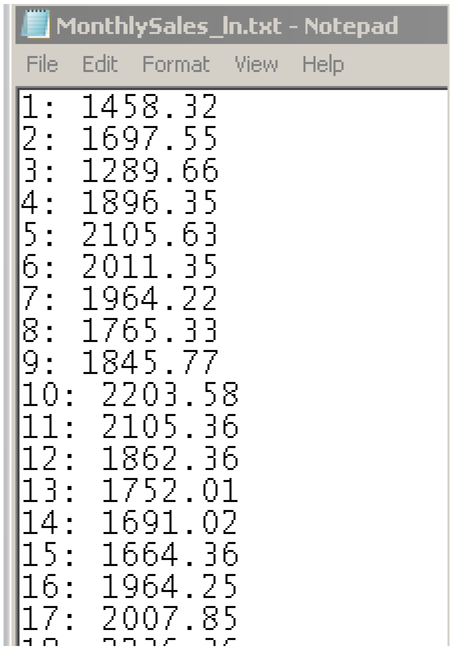


```
Please enter a positive integer: 6
#***********#
##*********##
###*******###
####*****####
#####***#####
######*######
```

Grading criteria:

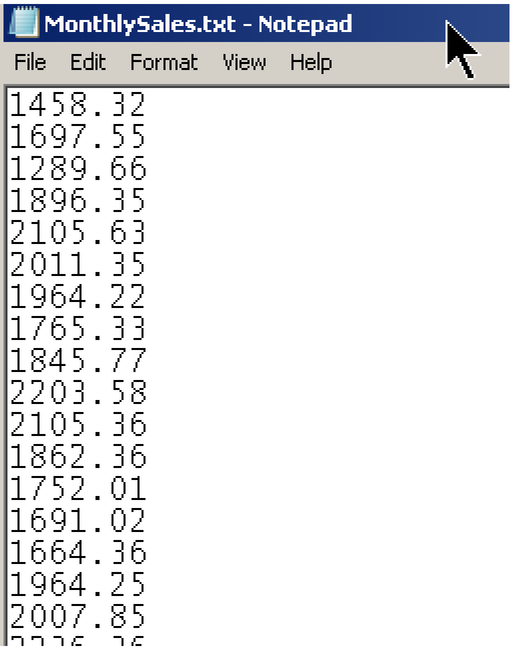
(1) Correctness (15 points total):
  (a) The code can be compiled without any syntax error
  (b) The code can generate the requested results
  (c) The program is documented using comments
(2) Technique used (15 points):
  (a) Either **`while`** loop, **`do while`** loop, or **`for`** loop should be used to complete the program.
  (b) **`if-else`** or **`switch`** statements should be used to complete the program.

3. (40 points) Write a program that would ask the user to enter an input file name, and an output file name. Then the program reads the content of the input file, and then writes the content of the input file to the output file with each line proceeded with a line number followed by a colon. The line numbering should start at 1.



```
1:  1458.32
2:  1697.55
3:  1289.66
4:  1896.35
5:  2105.63
6:  2011.35
7:  1964.22
8:  1765.33
9:  1845.77
10:  2203.58
11:  2105.36
12:  1862.36
13:  1752.01
14:  1691.02
15:  1664.36
16:  1964.25
17:  2007.85
```

Output file with line number added.



```
1458.32
1697.55
1289.66
1896.35
2105.63
2011.35
1964.22
1765.33
1845.77
2203.58
2105.36
1862.36
1752.01
1691.02
1664.36
1964.25
2007.85
```

Original input file

Grading criteria:

(1) Correctness (20 points total):
  (c) The code can be compiled without any syntax error
  (d) The code can generate the requested results
  (e) The program is documented using comments
(3) Technique used (20 points):
  (a) **`while`** loop should be used to complete the program.
  (b) **`Scanne`**r class is used
  (c) **`PrintWriter`** class is used
  (d) The files are closed before the program terminates.