# COSC 1436: Programming Fundamentals I

# (Assignment 2)

* **DUE Time and Date**: **11:59pm, Friday, 10/12/2018**
* **What to submit:** **LastNameFirstNameCarSalesCommission.java**

**input.txt** **LastNameFirstNameFileInputOutput.java**

**Note 1:** Unless otherwise mentioned, you are asked to upload ONLY your **java source files** (not the class files) through blackboard. Email submission is not accepted, because of confusion in grading.

**Note 2:** If your programs contain any syntactical errors, no points will be given. Thus, please make sure your programs are properly compiled with computers at the CS labs, not only in your laptop or desktop environments.

**Note 3:** No late submission will be accepted, thus keep the deadline.

**Note 4:** Grading will be divided into two categories, formatting and logic, where formatting compromise 25% of your total grade. Formatting will be based on the following rules.

**Rule 1:** Naming is an important issue in Java. Not only you need to define meaningful variable names, but also have to give appropriate names for the physical java file, which should be the same as your public class name that you edit.

Unless otherwise mentioned, you will follow **the industry standard for Java naming convention**:

(1) Java Classes start in uppercase and each individual word in the class name is capitalized;

(2) All Java methods and variables start in lowercase and each individual word in the method and variable is capitalized;

(3) Each final variable (known as a constant) should be written in all uppercase.

**Rule 2:** There should be a space around all operators (e.g., 3 + 5, not 3+5). In addition, spacing with regards to parentheses should be consistent.

**Rule 3:** In addition to the Java naming convention, you are asked to add your name in front of each class name like **LastNameFirstNameClassName.java.**

For instance, if your name is “John Doe” and the class name is “RightTriangle”, then your class name in your source code should be “DoeJohnRightTriangle” and your corresponding physical file name should be “DoeJohnRightTriangle.java”.

**Rule 4:** Everything nested inside of an open brace should be indented with regular-sized spaces (say, 4 or 8 spaces). The open brace for functions and classes should (1) come at the end of the line and be preceded by a space like

**public class DoeJohnRightTriangle {**

**public static void main() {**

**}**

**}**

or (2)start with the new line as shown below:

**public class DoeJohnRightTriangle**

**{**

**public static void main()**

**{**

**}**

**}**

**Rule 5:** Always type block comments to include title of the project, program’s purpose, your name, the date, and the version number as in the lectures or in the labs. For example,

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**@Title: LastNameFirstNameClassName**

**@Purpose: To verify the edit, compile, execute function in Textpad**

**@Author: (your last & first name)**

**@Date: (today’s date)**

**@Version: 1.0**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**Or**

**/\*\***

**@Title: LastNameFirstNameClassName**

**@Purpose: To verify the edit, compile, execute function in Textpad**

**@Author: (your last & first name)**

**@Date: (today’s date)**

**@Version: 1.0**

**\*/**

## Question 1 (60 points): Car Sales Commission

Write a program to calculate the commission a sales person earned at a local car dealership. The following is the rules how the commission of a sales person is calculated.

Each sales person has his base salary $3,000.

The dealership has 3 types of vehicles: sedan, truck and SUV. A sales person earns commission based on the quantity he sold for each type of vehicle. The commission is shown in the table below.

For the same type of vehicles, If a sales person sold 0 – 10 (including 10) vehicles, he would get level 1 commission for each vehicle he sold. If he sold 10 – 20 (including 20), he would get level 2 commission for the 11th – 20th vehicle he sold. If he sold more than 20, he would get level 3 commission for the vehicles starting from the 21th.

For example, if a sales person sells 23 sedans, he will get:

Base salary + (level 1 commission \* 10) + (level 2 commission \* 10) + (level 3 commission \* 3)

= $3000 + $80 \* 10 + \*120 \* 10 + $160 \* 3

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1. Sedan | 1. Truck | 1. SUV |
| 1. level 1 commission | 1. $80 | 1. $100 | 1. $100 |
| 1. level 2 commission | 1. $120 | 1. $140 | 1. $150 |
| 1. level 3 commission | 1. $160 | 1. $180 | 1. $200 |

Write a Java program to calculate the salary of a sales person at the dealership. The program will prompt the use for the following input: the sales person’s name, the number of sedan he sold, the number of truck he sold, and the number of SUV he sold. The program will then calculate the salary for the person, and display it on the screen. You program should work for as many person as needed.

A sample code execution should be as following:

Please enter your name: John

How many sedans you have sold this month: 9

You commission for the sedan sale is:

$80 \* 9 = $720

How many trucks you have sold this month: 15

You commission for the truck sale is:

$100 \* 10 + $140 \* 5 = $1700

How many trucks you have sold this month: 3

You commission for the SUV sale is:

$100 \* 3 = $300

John, you salary of this month is:

$3000 + $720 + $1700 + $300 = $5720

Do you want to calculate salary for another person? Type Y for yes, others for no: N

Thanks for using this system. Bye.

The class that you will be writing will be called **LastNameFirstNameCarSalesCommission.java**.

**You are required to use loop and conditional statements.** Grading criteria also include good documentation, descriptive variable names, and adherence to the coding convention noted on pages 1 & 2.

Your file will have the following documentation comments before the class header:

1. **/\*\***
2. @Title: LastNameFirstNameCarSalesCommission
3. @Purpose: To practice java loop and conditional statements.
4. @Author: (your last first name)
5. @Date: (today’s date)
6. @Version: 1.0
7. \*/

## Question 2 (40 points): File input and output

Write a program to:

1. Read a file which contains a list of students’ name ;
2. Output the two-letter initial, both in capital letter, to an output file output.txt;

Be sure to test to see if you opened the input and output files before you use them.

**You are required to use file input and output class.** Grading criteria also include good documentation, descriptive variable names, and adherence to the coding convention noted on pages 1 & 2.

You will first create a name.txt file, and input the following 10 randomly generated names:

Lilliana Sheley

Vincenza Teamer

Theron Beckler

Kacie Velazquez

Ione Exum

Mel Lade

Fernando Boxx

Leonila Discher

Sophie Pospisil

Verlene Knapik

Your program will then read each line, and find the initial for the person’s first and last name, and output the first and last name to the output.txt. You output.txt should like the same:

LS

VT

TB

…

Hint: the initial of the first name is the first letter. Use the indexOf() method of the String class to find the index of the blank space, the initial of the last name is the next index after the space. Assume each name takes one line, and first and last name are separated by one space.

Your file will have the following documentation comments before the class header:

1. **/\*\***
2. @Title: LastNameFirstNameFileInputOutput
3. @Purpose: To practice java File input and output.
4. @Author: (your last first name)
5. @Date: (today’s date)
6. @Version: 1.0
7. \*/