

METRO STATE UNIVERSITY

ICS 141 - 02: Problem Solving with Programming
Spring 2023

Assignment 1: Problem solving using variables and I/O

Out: Wednesday, January 11th, 2023

Due: Wednesday, January 25th, 2023

Total points: 25

Objective

You are asked to write a program to help a small company calculate the amount of money to pay their employees. In this simplistic world, the company has exactly three employees. However, the number of hours per employee may vary. The company will apply the same tax rate to every employee. To receive full credits, create a video recording less than 10 minutes with your **face on camera** describing your code step by step. Upload the java package as well.

Problem Description

Inputs (entered by user of the program)

- Name of first employee
- Hourly rate
- Number of hours worked

- Name of second employee
- Hourly rate
- Number of hours worked

- Name of third employee
- Hourly rate
- Number of hours worked

- Tax rate (between 0 and 1.0)

Processing and Output

- Calculate and display the amount each employee will be paid before taxes
- Calculate and display the amount each employee will be taxed
- Calculate and display the amount each employee will be paid after taxes have been withheld
- Calculate and display the total amount of taxes the company will withhold
- Display your name

You may separate the processing/calculation step from the output step or you may combine those steps. It is your choice.

Important Note: You may face a problem with `nextLine()` method does not stop to allow you to enter a text from the keyboard. This problem may appear when you call `nextLine()` after calling `nextInt()` or `nextDouble()`. The reason for this problem is that the `nextLine()` reads the new line (i.e., enter) that you type after writing your integer or double from the keyboard. To solve this problem, add a redundant `nextLine()` method after each `nextInt()` or `nextDouble()`. For example, instead of using the following code to read an integer:

```
int x = scan.nextInt()
```

use the following instead:

```
int x = scan.nextInt()  
scan.nextLine() //redundant
```

Requirements

- Prompt the user for the inputs and store the values in variables
- You must include all the inputs and outputs listed above and perform the calculations correctly
- Make the output look attractive
- The program should display your name
- Comment your code. At the top of the program include your name, a brief description of the program and what it does and the due date.
- The program must be written in Java and submitted via D2L.

Grading

Your grade in this assignment depends on the following:

- Your submission meets all the requirements as described above.
- The program is robust with no runtime errors or problems.
- You follow the good programming practices as discussed in class (check Lecture 1's slides)
- You follow the below submission instructions.

Submission Instructions

- Follow the following steps to upload your code to D2L:
 - Create a java project and call it <yourLastName>Assignment1 (e.g., mine will be called DillonAssignment1)
 - Create one .java files to solve the problem described above. ○ Archive your .java file into a **zip** file using Eclipse using the following steps:
 - ✦ In Eclipse Project Explorer, right click on the `src` folder of the project and click on `Export`.
 - ✦ Choose `General` then `Archive File` and click `Next`.

- ✦ Use the `Browse` key to choose a folder to store the archive file on your hard drive and give the file the same name as your project (e.g., `Dillon1.zip`), then click `Save`, then click `Finish`.
- ✦ Upload the **.zip** file you created to the D2L folder called `Assignment 1`.
- ✦ It is important that you upload only **one** zip file. Your assignment will not be graded if you upload individual .java files to the drop box.