

METRO STATE UNIVERSITY

ICS 141 - 02: Problem Solving with Programming
Spring 2023

Assignment 2: Classes and Objects

Out: Wednesday, January 25th, 2023

Due: Wednesday, February 15th, 2023

Total points: 25

Objective

This assignment builds on Assignment 1. Follow the instructions below. Update your existing code and add each of the requirements to complete the assignment. To receive full credits, create a video recording less than 10 minutes with your **face on camera** describing the additional changes to the code. Also upload the java source code.

Part 1. Declare 'Employee'

In this Assignment, you will implement a Java class, called '**Employee**', which can be used to represent employee objects in an application to be used, for example, in an HR application. You will then test your class by implementing another class, called '**EmployeeDriver**'.

To start, Open Eclipse and create a new project and name it 'HRApplication' then follow the following steps:

Create a class called 'Employee' inside the 'HRApplication' project.

1. Declare the following instance variables in 'Employee'. Remember that instance variables must be declared as private.
 - a. **name** (of type String).
 - b. **hourlyRate** (of type double).
 - c. **hours** (of type int).
2. Declare the following variable to be a Constant.
 - a. **TAX** (of type double). Tax rate (between 0 and 1.0)

Part 2. Constructor and Getters/Setters

1. Declare a constructor for the 'Employee' class that takes **three input** parameters and use **the input parameters to initialize the instance variables**.
2. Create getters and setters for each instance variable.
3. Implement a '**toString**' method to print the employees' information if one doesn't exist.

Part 3. Declare an 'EmployeeDriver'

1. Go to the driver class, inside the **main** method, declare **a variable of type 'Employee'** and call it '**myEmployee**'.
2. Using a sentinel loop, prompt the user for a **name**, a **rate** and **hour(s)** worked. Use the inputs to instantiate an employee object and to initialize the instance variables. Choose any values. (Note: you have to use **new** keyword to instantiate variables in the employee class.

3. Inside the **while loop** check if the inputs, **rate** and **hours** are invalid; **less than 0**. Display an **error** message to the user if the values are invalid. Then prompt the user if they would like to continue or not.
4. Call the **'toString'** method from 'EmployeeDriver' to print the employees' information.
5. Test different inputs values to verify the code works as expected and that it executes error free

Part 4. Methods – Do not start until after Feb. 1st lecture

1. Create a method with a return type double to calculate and display the amount each employee will be paid before taxes.
2. Create a method with a return type double to calculate and display the amount each employee will be taxed
3. Create a method with a return type double to calculate and display the amount each employee will be paid after taxes have been withheld
4. Create a method with a return type double to calculate and display the total amount of taxes the company will withhold
5. Call/Invoke each method from the Driver class on the employee object and display the results.

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. o 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.