COP 3330- **HW2**

In Java, Implement a simple weekly payroll program. The input will consist of the employee’s full name, employee’s ID (String), the number of hours worked per week (double), the hourly rate (double), the income tax is 6%. Your program should calculate the total gross pay, and the net pay and display a paycheck similar to:

---------------------------------------------------------------------------

Employee’s name: John Smith

Employee’s number: js1200

Hourly rate of pay: 10.50

Hours worked: 36.00

Total Gross Pay: $378.00

Deductions

Tax (6 %): $22.68

Net Pay: 355.32 Dollars

---------------------------------------------------------------------------

In the rest of this HW, Salary means the Gross Pay= hoursWorked times hourlyRate

Your program must include the class **Employee** whose private fields are:

* **fullName**: String
* **employeeNumber**: String
* **payRate**: double
* **hoursWorked**: double

(Don’t add any fields to the class **Employee**)

In addition to that, equip the class Employee with these methods:

* **Employee (String fullName, String employeeNumber, double payRate, double hoursWorked)**

This constructor assigns the parametres fullName, employeeNumber, payrate, hoursWorked to the data members.

* For each of the private data members above, add a **Setter** and a **Getter**
* Override the **toString()** method to return the String of the form:

**[Employee Number/Full Name, x Hours @ y per hour]** where x is the number of hours worked and y is the pay rate. As an example,

**[js1200/John Smith, 36 Hours @ 10.5 per hour]**

* double netPay (), the **private** method that returns the net pay of the employee. The net pay is calculated by deducting 6% from the gross pay. See sample check above.
* void printCheck (), the method that prints the paycheck of the employee as shown above. The printCheck method calls the private netPay method to get the net pay of the employee. So do not recalculate net pay in this printCheck method

After completing the code of the class Employee, work on the class Company whose skeleton is provided below. Have all the three classes in one ***.java*** file (***DriverClass.java***) and submit!

What is being asked to code is marked in comments throughout the ***DriverClass.java*** file. See it below.

You may work on this assignment with one or two classmates (no more than 3 per group). **PLEASE** add your name and the names of your groupmates as a comment so the TA won’t have to grade the same code.

Note well that whether you work with a group of students or solo, you must submit your ***.java*** on Webcourses. (This is going to be the case for all the future HWs and the final project).

**One more thing…**

**If you want to ask someone to write the code for you, go for it!**

**If you want to add someone’s name to your code as a favor, go for it!**

**But we truly hope that you don’t. Be an honest learner!**

***All students must use DriverClass.java shown below.***

//Names of Students who worked together

**public** **class** DriverClass {

**public** **static** **void** main(String[] args) {

String fullName = "Erika T. Jones";

String employeeNumber = "ej789";

**double** payRate = 100.0, hoursWorked = 1.0;

// TA will change the payrate and the hours worked to test your code

Employee e;

e = **new** Employee(fullName, employeeNumber, payRate, hoursWorked);

System.***out***.println(e); // To Test your toString method

e.printCheck(); // This prints the check of Erika T. Jones

Company company = **new** Company();

company.hire ( **new** Employee ("Saeed Happy", "sh895" , 2 , 200) );

company.hire (e);

company.*printCompanyInfo*();

company.hire( **new** Employee("Enrico Torres" , "et897" , 3 , 150) );

//You may add as many employees to company as you want.

//The TAs will add their own employees

//Make sure that each employee of company has a unique employeeNumber

company.printCheck("ab784");

company.deleteEmployeesBySalary(256.36);

company.reverseEmployees();

System.***out***.println( company.SearchByName("WaLiD WiLLiAms") );

company.printEmployees();

System.***out***.println("Bye!");

}

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**class** Employee {

//Add the private attributes and the methods as mentioned above…

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**class** Company {

**private** ArrayList<Employee> employeeList;

**private** String *companyName*;

**private** **static** String *companyTaxId*;

//Add static Setters and Getters for companyTaxId. We assume that

//all companies share the same companyTaxId and that may change

//Add Setter and Getter for the companyName

//No need to add a Setter and Getter for employeeList

**public** Company() {

employeeList = **new** ArrayList<>();

*companyName* = "People's Place";

*companyTaxId* = "v1rtua7C0mpan1";

}

**public** **boolean** hire ( Employee employee ) {

//Add empoyee to employeeList

//Note well that we can't add an employee whose employeeNumber already //assigned to another employee. In that case, this method returns false.

//This method returns true otherwise

}

**public** **void** printCompanyInfo() {

//This method prints the compay name, the tax id and the current number //of employees

//You may choose to print that any way you like!

}

**public** **void** printEmployees() {

//This methods prints all employees (One employee per line)

//Note that you already have ***toString*** in Employye

}

**public** **int** countEmployees( **double** maxSalary ) {

//This method returns the number of employees paid less than maxSalary

}

**public** **boolean** SearchByName (String fullName ) {

//This method returns true if fullName exists as an employee.

//It returns false otherwise

//this is a not a case sensitive search.

}

**public** **void** reverseEmployees () {

//This method reverses the order in which the employees were added to //the list. The last employee is swapped with the first employee, the //second last with the second and so on..

}

**public** **void** deleteEmployeesBySalary (**double** targetSalary ) {

//This method deletes all employees who are paid targetSalary as a gross //salary

}

**public** **void** printCheck ( String employeeNumber) {

//This method prints the check of the employee whose employee number is //employeeNumber. It prints NO SUCH EMPLOYEE EXISTS if employeeNumber is //not a registered employee number.

}

}//end of class Company