**Unit 9-HW (Last HW)**

Create the class ***Employee*** with ***int id****,* ***String name****,* ***double salary*** and ***int numberOfDependents*** as ***private*** fields. Override the ***toString*** method to print an employee in the format [id,name,net salary], where:

Net salary = salary\*0.91 + (numberOfDependent \*0.01\*salary)

Add the constructor ***public Employee (String name, double salary, int numberOfDependent)*** to set this.name, this.salaray and this.numberOfDependent using the corresponding parameters, and set the id to be the same of the ascii codes of the string: ***name.toUpperCase().*** Here ***toUpperCase()*** is the String’s method that returns the string of all of the characters in in upper case. Consequently, ***scotty Riter*** and ***scottY riTer*** are the same. You may change the names to be all in capital lettersNames are not to be changed to upper case.

The goal of this HW is to create a ***special*** ***linked list*** to store the employees. We seek to implement a structure that can be pictured as:

***company***

***Dane Ali***

***Ed Renu***

***Rim Oz***

***Kim Oz***

***Aidan Jones***

***Nadia Jones***

***Naadi Jones***

The dots represent null references of type Employee. Each node of ***company*** should contain two references that you may call ***next*** and ***below***, in addition *to* ***Employee e*** . (The picture above shows the names of the employees only, but your code should store an object of type employee, not just the name). That is:

***class Node{***

***private Employee e;***

***private Node next;***

***private Node below;***

***}***

One way to construct such linked list, we use:

***class LinkedList {***

***Node company;***

***public LinkedList (){***

***company = null;***

***}***

***}***

Note that the employees ***Aidan Jones***, ***Nadia Jones*** and ***Naadi Jones*** have the same id (collision case here), and that is why you see ***Nadia Jones*** and ***Naadi Jones*** are inserted to the sub-linked list whose head if the ***below*** reference of ***Aidan Jones***.

So if we add ***Dean Ali***, the list above becomes:

***company***

***Dane Ali***

***Ed Renu***

***Rim Oz***

***Kim Oz***

***Aidan Jones***

***Nadia Jones***

***Dean Ali***

***Naadi Jones***

The names of the list above could have been added in this order:

Kim Oz, Rim Oz, Dane Ali, Aidan Jones, Nadia Jones, Ed Renu, Naddi Jones and Dean Ali.

Your code should provide the code that allows to:

1. Print all the employees ([id,name,net salary])
2. Add a new employee
3. Search for an employee by name.
4. Find the highest net salary (Just the number)
5. Delete an employee by name

The Driver class (DriverClass.java) is provided below. Don’t change anything in the DriverClass.java

(Once again, DriverClass.java should contains all the classes of this HW, not just the DriverClass )

**public class DriverClass {**

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

**LinkedList list = new LinkedList();**

**list.addNewEmployee(new Employee ("Kim Oz", 1235.5, 3));**

**list.addNewEmployee(new Employee ("Rim Oz", 8235.5, 1));**

**list.addNewEmployee(new Employee ("Dane Ali ", 3235.5, 0));**

**list.addNewEmployee(new Employee ("Aidan Jones ", 2035.5, 2));**

**list.addNewEmployee(new Employee ("Nadia Jones", 5035.5, 3));**

**list.addNewEmployee(new Employee ("Ed Renu", 6035, 2));**

**list.addNewEmployee(new Employee ("Naadi Jones", 36035.75, 5));**

**//The TAs may use less or more names.**

**list.printAllEmployees();**

**System.out.println("The highest net salary = " list.highestNetSalary());**

**list.deleteEmployeeByName("Rim Oz");**

**list.deleteEmployeeByName("Nadia Jones");**

**System.out.println( list.searchByName("Gary D. Richardson") );**

**list.printAllEmployees();**

**}**

**}//end of DriverClass**

**//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**class LinkedList{**

**Node company;**

**public LinkedList () {**

**company = null;**

**}**

**public void printAllEmployees () {**

**//...**

**}**

**public void addNewEmployee (Employee e) {**

**//....**

**}**

**public boolean searchByName (String name) {**

**//...**

**}**

**public double highestNetSalary () {**

**//...**

**}**

**public void deleteEmployeeByName (String name) {**

**//...**

**}**

**}**

**//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**class Employee {**

**private String name; //Keep these fields private!**

**private int id;**

**private int numberOfDependent;**

**private double salary;**

**@Override**

**public String toString () {**

**return "...";**

**}**

**//...**

**}**

**//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**class Node {**

**private Employee e; //Keep these fields private!**

**private Node next;**

**private Node below;**

**//...**

**}**