CS 570: Programming Foundations
Fall 2017

Homework Assignment #8

Due date: Wednesday, December 6, 2017 by 11:59 PM.

Note: This and all assignments given in this course can be and must be solved using **only** the materials that have been discussed in class. Do not look for alternative methods that have not been covered as part of this course.

Program (100 points):

In this assignment, we are going to practice using linked lists. To that effect, we will create and manipulate a linked list of employees to simulate a payroll application. First, you will need to create an **`Employee`** class, according to the specifications below.

Specification for `Employee` class:

Attributes

- Employee ID: a string – the ID will contain digits and/or hyphens.
- Number of hours worked in a week: a double containing how many hours the employee has worked.
- Hourly pay rate: a double that represents how much the employee is paid for one hour or work.
- Gross wages: a double that stores the number of hours times the hourly rate.

Methods

- default constructor: initializes attributes to blank or zeroes.
- alternate constructor: receives and sets an employee ID.
- set methods for each attribute (accessors).
- get methods for each attribute (mutators).
- a printEmployee: that displays all the information of a given employee.

Driver File

When the program starts, it should ask the user to enter each of the employees' IDs. There should be no limit on the number of IDs the user can enter. The program then should display each employee identification number and ask the user to enter that employee's hours and pay rate. The program should calculate the gross wages for that employee (hours times pay rate). The data for all employees is stored in a linked lists of **`Employee`** nodes.

After the wages have been calculated for all the employees, the program should display each employee's identification number, hours worked, hourly rate, and gross wages.

Make sure your program validates the data input: do not accept negative numbers for hours worked or numbers less than 6.00 for pay rate.

Here are a couple of sample runs of a typical solution to this problem:

```
                ACME Payroll
----------------------------------------

Enter an employee ID or 000 to stop: 123
Enter an employee ID or 000 to stop: 456
Enter an employee ID or 000 to stop: 789
Enter an employee ID or 000 to stop: 000

Enter the total number of hours worked this week for employee 123: 20
Enter the total hourly rate ($6.00 or more): 15.50

Enter the total number of hours worked this week for employee 456: 15
Enter the total hourly rate ($6.00 or more): 6.00

Enter the total number of hours worked this week for employee 789: 36
Enter the total hourly rate ($6.00 or more): 20.50


Wages due this week
---------------------------------
ID      Hours   Rate      Wages
---------------------------------
123     20.00   $ 15.50  $  310.00
456     15.00   $  6.00  $   90.00
789     36.00   $ 20.50  $  738.00
```

```
                ACME Payroll
----------------------------------------

Enter an employee ID or 000 to stop: 1234
Enter an employee ID or 000 to stop: 5678
Enter an employee ID or 000 to stop: 1089
Enter an employee ID or 000 to stop: 7654
Enter an employee ID or 000 to stop: 1133
Enter an employee ID or 000 to stop: 000

Enter the total number of hours worked this week for employee 1234: -9
Negative values are not allowed. Try again.
Enter the total number of hours worked this week for employee 1234: 9
Enter the total hourly rate ($6.00 or more): 5.50
The hourly rate must be at least $6.00. Try again.
Enter the total hourly rate ($6.00 or more): 6.00

Enter the total number of hours worked this week for employee 5678: 12
Enter the total hourly rate ($6.00 or more): 11.75

Enter the total number of hours worked this week for employee 1089: 25
Enter the total hourly rate ($6.00 or more): 15.50

Enter the total number of hours worked this week for employee 7654: 15
Enter the total hourly rate ($6.00 or more): 8.00

Enter the total number of hours worked this week for employee 1133: 40
Enter the total hourly rate ($6.00 or more): 15.50


Wages due this week
---------------------------------
ID      Hours   Rate      Wages
---------------------------------
1234    9.00    $  6.00  $   54.00
5678    12.00   $ 11.75  $  141.00
1089    25.00   $ 15.50  $  387.50
7654    15.00   $  8.00  $  120.00
1133    40.00   $ 15.50  $  620.00
```

Note: Please make sure to submit well-written programs for these programming tasks. Good identifier names, useful comments, indentation, and spacing will be some of the criteria that will be used when grading this assignment.

Grading

Criteria for the Program	Points
Code Compiles Correctly – no compilation errors	20
Code runs properly on several tests and produces the correct results	20
Program Construction: all the required parts are there and logical flow is correct, including Employee class design and implementation, as well as the driver program	20
Correct implementation of linked lists: loops, iterators, methods.	20
User Interface: user knows what type of input the program expects and in what format it needs to be provided. Output is clear, well formatted, and easy to read/understand.	10
Code Style: good identifier names, named constants as needed, comments, indentation and spacing	10

How to submit your assignment

- Assignments must be submitted via Blackboard Learn.
  - Please note that assignments submitted via email will not be accepted.
  - Late assignments will not be accepted. Your work must be uploaded and submitted by 11:59 PM on the date it's due.

- For this assignment you must submit:
  - Two `.java` files for your source code: Employee class and the driver.
  - Do not submit files in any other formats – if you do, your assignment will not be graded.

Academic Honesty

You must be the sole original author of the solution you submit. You must compose all program and written material yourself. All material taken from outside sources must be appropriately cited.