CS 172 – Homework 4

Purpose:

After completing this assignment, you will have practiced using linked lists and linked list algorithms.

Description

For this assignment you will write a program to simulate a payroll application. To that effect you will also create an Employee class, according to the specifications below. Since an Employee list might be large, and individual Employee objects may contain significant information themselves, we store the Employee list as a **linked list**.

Specification for Employee class:

Attributes

- Employee ID: you can use a string – the ID will contain digits and/or hyphens. The ID must be provided during object construction and there should be no mechanism to change it later.
- Number of hours worked in a week: a floating-point number.
- Hourly pay rate: a floating-point number that represents how much the employee is paid for one hour of work.
- Gross wages: a floating-point number that stores the number of hours times the hourly rate.

Methods

- A constructor (__init__)
- Setter methods as needed.
- Getter methods as needed.
- This class should overload the __str__ or __repr__ methods so that Employee objects can be easily displayed using the print() function.

Specification for Node and LinkedList classes:

You will need to provide an implementation of the Node and the LinkedList classes. If you wish, you can re-use the code demoed during lecture but you will need to add a comment citing the source and you may need to modify it.

Script

Your main script will simulate a basic payroll system. There should be a menu with the following options:

- Add new employee: this option allows you to enter the ID of a new employee and his/her hourly rate, creates an `Employee` object accordingly, and adds it to the list of `Employees`.
- Enter Employee Hours: this option displays each employee ID and asks the user to enter the number of hours worked by each employee. The script should also (re)calculate the gross wages for each employee (hours times pay rate) and update their associated attributes accordingly.
- Display Payroll: this option displays each employee's identification number, hours worked, hourly rate, and gross wages.
- Update Hourly Rate: this option allows the user update the hourly rate of one employee. The user should enter the ID of the employee. If the ID exits in the list, then the hourly rate can be updated.
- Remove Employee for payroll: this option allows the user to remove an employee. The user should enter the ID of the employee. If the ID exits in the list, then the employee can be removed from the list.
- Quit the program

You should continue to display the choices, and respond to user input, until the user chooses to quit/exit the program.

Validation:

- Each employee number should be unique. Do not accept duplicate employee numbers.
- Do not accept negative numbers for hours worked.
- Do not accept numbers less than 6.00 for pay rate.

NOTE: Please keep in mind that you are expected to write a good quality, well formatted program. That means:

- Your program must have a header comment listing your full name, Drexel user id, and the purpose of the file – at the very list.
- User input must be validated and your program gracefully handle invalid inputs.
- Repetitive code (code that appears in multiple places in the main script) should be written as a function.
- Your program must use good style, including proper identifier names, useful comments, and proper use of indentation and whitespace.
- You program should also have an appropriate user interface so that anyone one using the program knows what to do and what to expect.

Grading

Criteria	**Points**
Employee class: `__init__` getter and setter methods, and `__str__` or `__repr__` overloaded method	15
`Node` class implementation	10
`LinkedList` class implementation	10
Main script: properly instantiates and handles a linked lists and node objects	15
Main script: validates user's input: hours worked, hourly rate, unique employee id	15
Main script: all the required parts are there: all menu options are handled correctly	20
Code follows good style guidelines and separate repetitive code into functions	10
All files have a header comment with your name and user id	5
Total possible points	**100**

NOTE: If you code has any runtime errors a 50-point deduction will be taken. Only portions of the code that execute without errors will be graded. If your script cannot run at all, you will receive 0 points.

How to Submit your assignment:

- Assignments must be submitted via Blackboard Learn.
    - Please note that assignments submitted via email will not be accepted.
    - Any late submissions will receive a -1% penalty per hour, up to 48hrs (after which, they will not be accepted, and will receive a 0%).

- For this assignment, you must submit a single zip (such as HW4.zip) file that contains:
    - **`Node.py`** – the node class implementation
    - **`LikedList.py`** - the linked list class implementation
    - **`Employee.py`** - the employee class implementation
    - **`main.py`** - your main script

Academic Honesty

You must be the **sole original author** of the **entire solution** you submit. You must compose all program and written material yourself. All material taken from outside sources (e.g. textbooks, in class examples, labs, etc.) must be appropriately cited.

----------SAMPLE RUN OF PROGRAM IN THE NEXT PAGE ----------------

```
>>> main()

*** CS 172 Payroll Simulator ***
a. Add New Employee
b. Enter Hours Worked
c. Display Payroll
d. Update Employee Hourly Rate
e. Remove Employee from Payroll
f. Exit the program

Enter your choice: a
Enter the new employee ID: 123
Enter the hourly rate: 7.75

*** CS 172 Payroll Simulator ***
a. Add New Employee
b. Enter Hours Worked
c. Display Payroll
d. Update Employee Hourly Rate
e. Remove Employee from Payroll
f. Exit the program

Enter your choice: a
Enter the new employee ID: 234
Enter the hourly rate: 13.50

*** CS 172 Payroll Simulator ***
a. Add New Employee
b. Enter Hours Worked
c. Display Payroll
d. Update Employee Hourly Rate
e. Remove Employee from Payroll
f. Exit the program

Enter your choice: c

 *** Payroll ***
Employee ID:  123
Hourly Rate:  7.75
Hours Worked: 0.0
Gross Wages:  0.0

Employee ID:  234
Hourly Rate:  13.5
Hours Worked: 0.0
Gross Wages:  0.0


*** CS 172 Payroll Simulator ***
a. Add New Employee
b. Enter Hours Worked
c. Display Payroll
d. Update Employee Hourly Rate
e. Remove Employee from Payroll
f. Exit the program
```

```
Enter your choice: b
Enter hours worked for employee 123 : 15
Enter hours worked for employee 234 : 11.50

*** CS 172 Payroll Simulator ***
a. Add New Employee
b. Enter Hours Worked
c. Display Payroll
d. Update Employee Hourly Rate
e. Remove Employee from Payroll
f. Exit the program

Enter your choice: c

 *** Payroll ***
Employee ID:  123
Hourly Rate:  7.75
Hours Worked: 15.0
Gross Wages:  116.25

Employee ID:  234
Hourly Rate:  13.5
Hours Worked: 11.5
Gross Wages:  155.25


*** CS 172 Payroll Simulator ***
a. Add New Employee
b. Enter Hours Worked
c. Display Payroll
d. Update Employee Hourly Rate
e. Remove Employee from Payroll
f. Exit the program

Enter your choice: e
Enter the ID of the employee to remove from payroll: 123
Done.


*** CS 172 Payroll Simulator ***
a. Add New Employee
b. Enter Hours Worked
c. Display Payroll
d. Update Employee Hourly Rate
e. Remove Employee from Payroll
f. Exit the program

Enter your choice: f
Good-Bye!
```