CST8132 Object-Oriented Programming

Lab 4 Abstract Class: LMS - Paying Librarians

Due Date: Week 5 – in own lab hours

Marks: 10 marks (worth 4% of term mark)

Demo: Demo your code and output to your lab professor during your own lab hours.

Recommended Reading: Chapter 9 & 10 of Deitel and Deitel, Java How to Program book

Video Resource: Watch this and this. Links also posted at the end of Week 3 learning content.

Instructions/Requirements

In this lab, we are writing an object-oriented program incorporating abstraction, encapsulation, polymorphism and inheritance that uses an array of Librarian as an attribute of a Payroll object with two of each of the following kinds of librarians:

- **Parttime**
  - First name
  - Last name
  - email
  - hourly rate: $45 (same for all parttime librarians)
  - hours per week
  - tax rate: 15% (that is, 0.15) (only contractors have this property)
  - method of payment: mail a check
  - weekly payroll calculation: hours * rate * (1 - tax rate) + (a bonus: $10 * a quarter of hours worked each week).

- **Fulltime**

  - First name
  - Last name
  - email
  - hourly rate: $30 (same for all fulltime librarians)
  - hours per week
  - deduction rate: 17% (that is,0.17) (only regular librarians have this property)
  - method of payment: direct deposit
  - weekly payroll calculation: hours * rate * (1 - deduction rate) + (a bonus: $5 * a fifth of hours worked each week).

You may start with your lab 3 program. To do this it is important that move the details of worker class to Librarian class and delete Worker class. Payroll class will replace library class as you will see below.

Write down Java classes including an **abstract** Librarian class with two abstract methods - one that does the payroll calculation and the other that prints the method of payment. Consider first & last name, email, hourly rate, hours per week, tax rate and deduction rate as **attributes**, and consider the method of payment and payroll calculation to be **behaviors**.

Implement the method of payment behavior by printing out a suitable string when the payment is made. For example, your payment method could take an amount parameter, and consist of a single line:
System.out.println("Direct depositing " + amount + " to " + name + "'s bank account");

Your method for calculating the payment should return a double.

The constructor for your Parttime and Fulltime classes will accept first name, last name, email and number of hours worked as parameters, and initialize all those properties. Properties like hourly rate, tax rate and deduction rate will be initialized with the values given above. *Super call must be made to call super class constructor from the child class constructor.*

Here is the driver class:

```
package payrollabstractclass;

import java.util.Scanner;

public class Lab4Test {

	public static void main(String[] args) {
		//creating an instance for Scanner class
		Scanner input = new Scanner(System.in);
		System.out.println("\nEnter number of librarians: ");
		int num = input.nextInt();
		//creating payroll object for Payroll class with num parameter
		Payroll payroll = new Payroll(num);
		//invoking readLibrarians method
		payroll.readLibrarians();
		System.out.println("\n\nSummary of Payroll Processing");
		System.out.println("================================");
		//invoking processPayroll method
		payroll.processPayroll();
		input.close();
	}

}
```

In Payroll class, you need to read personal information including the type of the librarian. For the payroll calculation, for each librarian, calculate the amount owed to that librarian and pass that as a parameter to the librarian's payment method. You need to write the code that performs the calculation (**do not** hard-code the payroll figures).

NOTE: Your output will display different payroll figures that include the bonus in payroll calculations.

Grading Scheme

Item	**Marks**
Librarian class (correct access specifiers, constructors, 2 abstract methods)	2
Fulltime class (correct access specifiers, 2 overridden methods)	2
Parttime class (correct access specifiers, 2 overridden methods)	2
Payroll class (correct access specifiers, constructors, 2 methods)	2
Comments (class header, wherever required, provide comments)	2

Submission

Submit a zip folder of source files here.

Assessment : Grade is obtained only after the demo during the scheduled lab:

Demonstrate your work to your lab professor.

Looking forward
Once you are done with this lab, think about interfaces.

Sample output (User input in blue)

```
Enter number of librarians: 4

Enter details of librarian 1
============================
Enter first name: James
Enter last name: Martin
Enter email: jmarting@gmail.com
Hours worked: 37.5
1. Fulltime
2. Parttime
Enter type: 1

Enter details of librarian 2
============================
Enter first name: Mary
Enter last name: Hansey
Enter email: mhansey@yahoo.com
Hours worked: 42.5
1. Fulltime
2. Parttime
Enter type: 2

Enter details of librarian 3
============================
Enter first name: Kennedy
Enter last name: Lovet
Enter email: kenlovet@gmail.com
Hours worked: 30
1. Fulltime
2. Parttime
Enter type: 1

Enter details of librarian 4
============================
Enter first name: Reuben
Enter last name: Harrets
Enter email: rharrets@gmail.com
Hours worked: 28.75
```

1. Fulltime
2. Parttime

Enter type: 2

Summary of Payroll Processing
==========
Direct deposit of $933.75 to James Martin's bank account
Mailed a cheque to Mary Hansey for $2199.375
Direct deposit of $747.0 to Kennedy Lovet's bank account
Mailed a cheque to Reuben Harrets for $1487.81