Lab 3: Library Management System - (Inheritance and Polymorphism)

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

Grading: Demo your code and output to your lab professor during your lab session.

Useful Reading: Chapter 9 of Deitel and Deitel, Java How to Program book

Objectives

We will employ the OOP concepts of inheritance and polymorphism to create a more functional Library Management system by creating the listed classes as per the instructions given below. A library has different types of workers who include librarians and cleaners in addition to users in their system. In this lab, we are creating these classes - Worker, Librarian, Fulltime, Parttime and Library. All worker data fields like first name, last name, email, phone number should be in Worker class. Librarian class should extend Worker class. It should have a librarian ID attribute. As we know that the library has fulltime and parttime librarians, we need to create two classes – Fulltime and Parttime, both extends Librarian class. Fulltime class should have a salary attribute that represents the monthly salary. Parttime class should have attributes to represent hourly rate and number of hours worked. Library class will manipulate the library librarians. More classes like Client/User, different types of Client/User classes etc. will be added later. Note: Unlike previous lab, data will entered through a scanner instead of initializing it with a constructor. Remember to close your scanner(s) to prevent memory leaks.

As the first step, you need to create the following classes:

`Worker` class

Instance variables: `firstName`(String), `lastName`(String), `email`(String), `phoneNumber` (long). Librarian class will be inheriting this class. So, think about the access specifiers for the instance variables.

Constructor: parameterized constructor that gets values to set all properties of a person (It's upon you to think and decide if you need a no-arg constructor).

Methods: getter to return name. Name should be returned as one string. If your first name is "Peter" and your last name is "Ken", getter should return "Peter Ken".

Librarian class (inherits from class Worker)

Instance variables: `librarianID` (int)

Constructor: parameterized constructor that initializes a librarian with librarian ID and all worker properties. As you have a parameterized constructor for Worker class, use that to set worker properties.

Methods:

1. `readLibrarian()`: accepts nothing, returns nothing. Reads all librarian information.
2. `printLibrarian()`: accepts nothing, returns nothing. This method prints details of a librarian using formatted output (use printf).

Fulltime class (inherits from class Librarian)

Instance variables: salary (double)

Methods:

1. readLibrarian(): accepts nothing, returns nothing. Make a call to the readLibrarian() method of the parent class. Then, reads annual salary from the user, converts it to monthly salary and save it in salary instance variable.
2. printLibrarian(): accepts nothing, returns nothing. Make a call to the printLibrarian() of the parent class. Then, prints salary info (formatted output).

Parttime class (inherits from class Librarian)

Instance variables: hourlyRate (double), numHours (double)

Methods:

1. readLibrarian(): accepts nothing, returns nothing. Make a call to the readLibrarian() method of the parent class. Then, reads hourly rate and number of hours worked.
2. printLibrarian(): accepts nothing, returns nothing. Make a call to the printLibrarian() of the parent class. Then, prints salary, which is the product of hourly rate and the number of hours worked (formatted output).

Library class

Instance variables: an array of Librarian named librarians

Constructor: parameterized constructor that creates the array of librarians with the given size (this size will be read in main(), and will be sent here when creating the Library object)

Methods:

1. readLibrarianDetails(): accepts nothing, returns nothing. In a for loop, read details of all librarians. First, read the type of the librarian. Based on the type of the librarian, corresponding array object needs to be created (Polymorphism). Then, call readLibrarian() method.
2. printLibrarianDetails(): accepts nothing, returns nothing. In a for loop, call printLibrarian() to print details of all librarians.
3. printLine(): *static* method that prints a line using "="
4. printTitle(): *static* method that prints the title of the output. This method gets the name of the library as a parameter, which will be used in the formatted print statement. printLine() method will be called from this method to print lines.

Lab3Test class

This is the driver class (test class), which means this class has the main method.

Main method

- This method read the name of the library (example: "Quality") and the number of librarians (saved in num).
- A Library object will be created with the 'num'.
- Call readLibrarianDetails() method to read details of all librarians
- Print the title and the header row
- Call printLibrarian() method to print details of all librarians.

Format your code with proper indentation and formatting. Your code should be properly commented. Test plan and external documentation are not required for this exercise, but in future labs they will be required. Submit here or through Activities tab =>Assignments=>Lab3.

Grading Scheme

Item	Marks
Worker class (correct access specifiers, constructors, 1 method)	1
Librarian class (correct access specifiers, constructors, 2 methods)	1
Fulltime class (correct access specifiers, 2 methods)	1
Parttime class (correct access specifiers, 2 methods)	1
Library class(correct access specifiers, constructors, 4 methods)	2

Comments (class header, wherever required, provide clear comments)	2
Properly formatted output (see expected output on page 3)	2

Submission

Demonstrate your work to your lab professor.

Getting ready for next lab

Once you are done with this lab, think about abstraction.

Expected Output (green – user input)

```
Enter name of the library: OOP
How many librarians do you have? 3
Enter details of librarian 1
   1. Fulltime
   2. Parttime
Enter type of librarian: 1
Enter Librarian ID: 110210
Enter first name: Peter
Enter last name: Ken
Enter email: ken@oop.com
Enter phone number: 123456789
Enter annual salary: 98000
Enter details of librarian 2
   1. Fulltime
   2. Parttime
Enter type of librarian: 2
Enter Librarian ID: 110220
Enter first name: Eric
Enter last name: James
Enter email: eric@oop.com
Enter phone number: 213546879
Enter hourly rate: 75
Enter number of hours worked: 5
Enter details of librarian 3
   1. Fulltime
   2. Parttime
Enter type of librarian: 1
Enter Librarian ID: 110230
Enter first name: Tim
Enter last name: Hortons
Enter email: tim@oop.com
Enter phone number: 321654987
Enter annual salary: 85450
================================================================================
                       OOP Library Management System
================================================================================
LibrarianID|                Name |          Email |        Phone |     Salary|
================================================================================
     110210 |           Peter Ken |    ken@oop.com |    123456789 |   8166.67 |
     110220 |          Eric James |   eric@oop.com |    213546879 |    375.00 |
     110230 |         Tim Hortons |    tim@oop.com |    321654987 |   7120.83 |
```