American University of Beirut

Department of Computer Science
CMPS 212- Summer 2021/22
Lab 02

Problem 1: Books & Authors

In this exercise you are asked to create three classes*: 1) date, 2) author and 3) book*. Details for each class is provided below.

1) ***Date class*** which contains:
   - Day, Month & Year
   - Setters and getters for each data member
   - the toString method that returns the date in the following format Date: 2/3/2022

2) ***Author class*** which contains:
   - First name, Last name, ***birthdate***, short biography and the number of published books
   - Setters and getters for each data member
   - the toString method that returns the author details in the following format:

   Author Details:
   Name: John Doe
   Birthdate: 1/2/1994
   Number of published books: 3
   Bio: John Doe is a professor and author at university X and the writer of the popular novel Y.

3) ***Book class*** which contains:
   - Book title, ***Author details***, ***published date*** , rating(over 5)
   - Setters and getters for each data member
   - the toString method that returns all the book details in a similar way to the author's toString
   - updatePublishedDate() and updateAuthor() which take care of updating the date and author fields respectively.

   Note: make sure to have no duplicate data member fields in the classes.

Problem 2: Employee Inheritance

Implement a class ***Employee*** with the following members:
- fields: int baseHours, double baseSalary, int baseVacationDays, string baseVacationForm
- Getters & setters for all fields

Implement a class ***Marketer*** that inherits ***Employee***.
- Marketers make $50,000 ($10,000 more than general employees)
- Write an additional method advertise() that prints "Act now, while supplies last!"

Implement a class ***Janitor*** that inherits ***Employee***.
- Janitors work twice as many hours per week as other employees (80 hours/week)
- Janitors make $30,000 ($10,000 less than general employees)
- Janitors get half as much vacation as other employees (only 5 days)
- Write an additional method clean() that prints "Workin' for the man".

Implement a class Lawyer that extends Employee.
- Lawyers vacation form is pink
- Lawyers have 5 additional vacation days
- Add a method sue() that prints "I'll see you in court!"

Implement a class ***HarvardLawyer*** that extends ***Lawyer***.
- Harvard Lawyers make 20% more money than a normal lawyer
- Harvard Lawyers get 3 additional days of vacations (i.e 5 + 3)

Write a driver code that tests all the classes and methods you defined and answer the following questions:
- Can we access a marketer's salary by defining a marketer's object?
- Can a Janitor advertise?
- Can a Harvard Lawyer sue?

Problem 3: Machine Class

Write a Java class called Machine which has:

* 2 instance variables brandName and power (to check if it is On or not)

* 2 constructors (a multi-argument constructor and a zero-argument one)

* getters for the two instance variables

* a setter for brandName

* 2 methods switchOn() and switchOff() which change the value of power

* a method called display() to print the status of the machine

Problem 4: WashingMachine Class

Write a Java class called WashingMachine which:

* is a subclass from Machine class

* has 2 new instance variables productionDate and capacity

* has 2 constructors (a multi-argument constructor and a zero-argument one)

* has getters and setters for the new instance variables

* has toString() method which calls the toString()of the super class

* has equals() method (check equality of all attributes except power)

* has compareTo() method to be used in sorting (according to brandName then capacity)

Problem 5: Inheritance Mystery

Assume that the following classes have been defined:

```
public class Pen extends Sock {
 public void method1() {
 System.out.print("pen 1 ");
 }
}
public class Lamp {
 public void method1() {
 System.out.print("lamp 1 ");
 }
 public void method2() {
 System.out.print("lamp 2 ");
 }
 public String toString() {
 return "lamp";
 }
}
public class Book extends Sock {
 public void method2() {
 System.out.print("book 2 ");
 super.method2();
 }
}
public class Sock extends Lamp {
 public void method1() {
 System.out.print("sock 1 ");
 method2();
 }
 public String toString() {
 return "sock";
 }
}
```

Given the classes above, what output is produced by the following code?

```
Lamp[] elements = {new Book(), new Pen(), new Lamp(), new Sock()};
for (int i = 0; i < elements.length; i++) {
 System.out.println(elements[i]);
 elements[i].method1();
 System.out.println();
 elements[i].method2();
 System.out.println();
 System.out.println();
}
```