

## Object-Oriented Programming Lab#7, Spring 2019

### Today's Topics

- Inheritance
- method override
- subclass polymorphism

### An Employee Record System

Create an Employee Record System for “UAP HR” department. Implement the system in such a way where user can 1) add new Employee to the system, 2) get the monthly salary of any employee, 3) increase the salary/rate of an employee [See Note\*\* below] and 4) display the list of Employee and their info. Each Employee is identified by **his/her name, employee id and position/designation**. There could be 3 types of employee;

- 1) Salaried employee -> this type of employees are paid a fixed monthly salary regardless of the number of hours worked.
- 2) Hourly employee -> They are paid by the hour. They have an hourly rate and their payment will depend on how many hours they worked. The more they work, the more they will be paid. So, the salary will be [hour worked per month\* hourly rate].
- 3) Commission employee-> They are paid a percentage of their sales. If their percentage(known as **commission**) is “a” and total monthly sale is “b”, the total monthly salary will be  $[a*b/100]$ ;

**Note\*\*:** For all 3 employees, **increase salary feature/functionality will increase the rate not the total salary**. Set the rate of each employee will do the following

- a. for **salaried** employee it would set the **monthly** rate,
- b. for **hourly** employee it will set the **hourly rate** and
- c. for **commission** employee it will set the **percentage**.

## Object-Oriented Programming Lab#7, Spring 2019

### ***What you need to do:***

- 1) Create an **Employee** class which has **3 instance variables; *name, id, designation***.
  - a. Create a **constructor** that takes initial value for those 3 attributes and initializes those attributes.Create the **following 3 methods as described**
  - a. ***void increaseSalary(double amt)***
    - Leave the method body empty.
  - b. ***double getSalary()***
    - This method returns 0(zero).
  - c. ***void display()***
    - This method displays the attributes in the format "Name:[name]; Id:[id]; Desig:[designation];".
- 2) Create **SalariedEmployee** class and make it the subclass of **Employee** class. Now do the following.
  - a. Add a new attribute name ***monthlySalary***.
  - b. Create a **constructor** that takes initial value for those 4 attributes and initializes those attributes.
    - Inside the constructor, call parent class's constructor. Also set the ***monthlySalary***.Override the **following 3 methods as described**
  - c. ***void increaseSalary(double amt)***
    - Inside the method, increase the ***monthlySalary*** by ***amt*** amount.
  - d. ***double getSalary()***
    - Returns the ***monthlySalary*** value.
  - e. ***void display()***
    - Call the ***display()*** method of parent class and then print "; Salary:[salary]".
- 3) Create **HourlyEmployee** class and make it the subclass of **Employee** class. Now do the following.
  - a. Add 2 new attributes name ***hourlyRate*** and ***hourWorked***.
  - b. Create a **constructor** that takes initial value for those 5 attributes and initializes those attributes.
    - Inside the constructor, call parent class's constructor. Also set the ***hourlyRate*** and ***hourWorked***.Override the **following 3 methods as described**
  - c. ***void increaseSalary(double amt)***
    - Inside the method, increase the ***hourlyRate*** by ***amt*** amount.

d. **double getSalary()**

- Returns the total payment/salary which will be [hour worked per month\* hourly rate].

e. **void display()**

- Call the **display()** method of parent class and then print “; Rate:[hourlyRate]”.

4) Create **CommissionEmployee** class and make it the subclass of **Employee** class. Now do the following.

a. Add 2 new attributes name **commission** and **sale**.

b. Create a **constructor** that takes initial value for those 5 attributes and initializes those attributes.

- Inside the constructor, call parent class's constructor. Also set the **commission** and **sale**.

Override the **following 3 methods** as described

c. **void increaseSalary(double amt)**

- Inside the method, increase the **commission** by **amt** amount.

d. **double getSalary()**

- Returns the total payment/salary which will be [**commission** \* **sale**].

e. **void display()**

- Call the **display()** method of parent class and then print “; Commission:[commission]”.

5) Now create another class **UapCSE** to represent the CSE department which has a list of Employee. So, there will be one attribute of type Employee Array or ArrayList to represent the list of the employee [**Employee[] employees**]. Add the following methods to this class.

a. **void addNewEmployee(Employee e)**

- Add the **Employee** e to **employees** array.

b. **void addNewEmployee(String n, String l, String d, double s)**

- Create a **SalariedEmployee** object using the parameter provided and add the object to **employees** array by calling the **addNewEmployee(Employee e)** method.

c. **void addNewEmployee(String n, String l, String d, double hr, int hw)**

- Create an **HourlyEmployee** object using the parameter provided and add the object to **employees** array by calling the **addNewEmployee(Employee e)** method.

d. **void addNewEmployee(String n, String l, String d, double p, double s)**

- Create an **CommissionEmployee** object using the parameter provided and add the object to **employees** array by calling the **addNewEmployee(Employee e)** method.

e. **Employee findEmployee(String id)**

- Loop through the **employees** and find the **Employee** whose id matches with the parameter provided.

f. **void increaseSalary(String id, double amt)**

- Find the Employee using the ***findEmployee*** method and call the ***increaseSalary*** method for that object.
- g. **double getSalary(String id)**
  - Find the Employee using the ***findEmployee*** method and call the ***updateSalary*** method for that object
- h. **void display()**
  - Loop though the ***employees*** and call display of Employee class for each item.

1) Create an **application class** (that has the main method) named “**UAP**” which will have the **main** method.

- In the main method, display the following menu to user and take necessary action.

- Input ‘1’ to add a new Employee.

You need to provide use a submenu to create different types of employee. So, you have to ask for appropriate input from the user depending on what type of employee(**Salaried, Hourly, Commission** ) they want to add and call respective ***addEmployee()*** method.

- Input ‘2’ to get Salary info of a specific Employee
- Input ‘3’ to increase the salary of an Employee.
- Input ‘4’ to display the list of the Employees.
- Input ‘0’ to exit the system.