**Object-Oriented Programming Lab#5, Fall 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 and 4) display the list of Employee and their info. Each Employee is identified by **his/her name, employee id, position/designation and salary.**

**Object-Oriented Programming Lab#5, Fall 2019**

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

1. Create an **Employee** class which has **4 private instance variables*; name, id, designation, salary***.
   1. Create a **constructor** that takes initial value for those 4 attributes and initializes those attributes.

Create the **following methods as** described

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

* Inside the method, increase the ***salary*** by ***amt*** amount.

1. ***String getId()***

* This method returns the id.

1. ***double getSalary()***

* This method returns the salary.

1. ***void display()***

* This method displays the attributes in the format “Name:[name]; Id:[id]; Designation:[designation]; Salary:[salary]”.

1. 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 **ArrayList** to represent the list of the employee [**ArrayList**<***Employee> employees***] and another attribute [name it as **name**] to store the name of the department.
   1. Create a constructor and pass department name as parameter.

* Inside the constructor initialize the ***name*** and instantiate the ***employees*** object.

Add the following methods to this class.

* 1. **void addNewEmployee(Employee e)**
* Add the **Employee** e to ***employees*** array.
  1. **void addNewEmployee(String nm, String id, String des, double sal)**
* Create an **Employee** object using the parameter provided and add the object to ***employees*** array by calling the **addNewEmployee(Employee e)** method.
  1. **Employee findEmployee(String id)**
* Loop though the ***employees*** and find the **Employee** whose id matches with the parameter provided. If the Employee is found in the list, return the **Employee**. Return **null** otherwise.
  1. **void increaseSalary(String id, double amt)**
* Find the Employee using the ***findEmployee(id)*** method. If the method returns an **Employee**, call the ***increaseSalary(double)*** method using that object.
  1. **double getSalary(String id)**
* Find the Employee using the ***findEmployee(id)*** method. If the method returns an **Employee**, call the ***getSalary()*** method using that object.
  1. **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.
   1. Inside the main method, create an object [name it ***myUap***] of **UapCse** class and then provide the following **menu** on the console. Once the user enters his/her option, you need to read the value and take appropriate action(See below) using the **myUap** object.
      * Input ‘1’ to add a new Employee.

If user chooses this option, you have to ask user for the employee name, id, designation and salary. After getting the value call **addEmployee(…)** method using **myUap** object.

* + - Input ‘2’ to get Salary info of a specific Employee

If user chooses this option, you have to ask user for the employee id. After getting the id, call **getSalary(…)** method using **myUap** object.

* + - Input ‘3’ to increase the salary of an Employee.

If user chooses this option, you have to ask user for the employee id. After getting the id, call **increaseSalary(…)** method using **myUap** object.

* + - Input ‘4’ to display the list of the Employees.

If user chooses this option, you have to ask user for the employee id. After getting the id, call **getSalary(…)** method using **myUap** object.

* + - Input ‘0’ to exit the system.

If user chooses this option, exit the system.