

Assignment10: Employee Compensation

CIS 210 – Fall 2016

Point: 100

Project Due: 11:59 PM on Thursday, 11/29/2016

Learning Objectives:

- ❖ To understand the use of Classes in Java
- ❖ To understand the concept of Class Inheritance, Polymorphism, and Interface
- ❖ To use Inheritance, Polymorphism, and Interface in Java
- ❖ To read in data from an existing file
- ❖ To sort data using selection sort algorithm

Problem Statement

- Design an **abstract** *Employee* class that has the following members:
 - A field for the ID of the employee (an integer)
 - A field for the year that the employee was hired (an integer)
 - A field for the base salary of the employee (double)
 - A field for the total compensation of the employee (double)
 - A constructor and appropriate getters and setters
 - A toString() method that displays the employee's name, the year in which the employee was hired, his or her base salary, and total compensation.
- Design a *JuniorEmployee* class that extends the *Employee* class. The *JuniorEmployee* class should have the following members:
 - A field for commission (double).
 - A constructor and appropriate setters and getters.
 - A toString() method that overrides the toString() method in the base class. The *JuniorEmployee* class's toString() method should display only the employee's ID, the year he or she was hired, and the commission.
- Design a *SeniorEmployee* class that extends the *Employee* class. The *SeniorEmployee* class should have the following members:
 - A field for the annual bonus (double).
 - A constructor and appropriate setters and getters.

- A toString() method that overrides the toString() method in the base class. The SeniorEmployee class's toString() method should display only the employee's ID, the year he or she was hired, and the employee's annual bonus.
- Design a *Manager* class that extends the *Employee* class. The *Manager* class should have the following members:
 - A field for stock dividend (double)
 - A constructor and appropriate setters and getters
 - A ShowDividend() method that displays the amount of dividend.
 - A toString() method that overrides the toString() method in the base class. The *Manager* class's toString() method should display only the employee's ID, the year he or she was hired, and the employee's dividend.
- Design an Interface named *EmployeeCompensation*. It has one methods: CalculateTotalCompensation() which returns the total compensation. *JuniorEmployee*, *SeniorEmployee*, and *Manager* classes MUST implement this Interface.
 - JuniorEmployee's total compensation is baseSalary + commission
 - SeniorEmployee's total compensation is baseSalary + annualBonus
 - Manager's total compensation is baseSalary + dividend

Demonstrate/test the classes in a program that has one Employee array. The program reads various employees' data from an existing file named "Employees.txt" and create appropriate (JuniorEmployee/SeniorEmployee/Manager) objects based the type information of an employee. For example, if an employee's type is Junior, the program should create a *JuniorEmployee* object. Then, insert various *JuniorEmployee*, *SeniorEmployee*, and *Manager* objects to the array elements.

1. The program uses *selection sort algorithm* to sort the array based on an Employee's ID in ascending order.
2. The program steps through the array, calling each object's toString() method.
3. The program steps through the array, calling the ShowDividend() method from every Manager object. Your program MUST be able to distinguish every Manager object from all of those Employee object in the array.

Input

The input is from an existing file named “Employees.txt”. Note, if the file does not exist, your program MUST display an error message and terminate gracefully.

Output

Your program should display the following on the console and to a file named *employeeOutput.txt*:

- Each employee’s ID, title (Senior, Junior, Manager), yearHired, and total compensation (base salary + commission/annual bonus/dividend), along with the content of its toString() method. They should be displayed in *ascending* order.
- For every manager, display his or her dividend using ShowDividend() method.

Sample output:

ID	YEAR_HIRED	TITLE	BASE_SALARY	COMPENSATION
11	1999	Senior	\$50,000	\$60,000
This is a senior employee. ID is 11, hired since 1999, and annual bonus is \$10,000.				
12	2000	Junior	\$50,000	\$55,000
This is a junior employee. ID is 12, hired since 2000, and commission is \$5,000.				
13	1980	Manager	\$80,000	\$100,000
This is a manager. ID is 13, hired since 1980, and stock dividend is \$20,000.				
14	2012	Junior	\$50,000	\$51,000
This is a junior employee. ID is 14, hired since 2012, and commission is \$1,000.				

Employee 13 is a manager. Dividend is \$20,000!

Good bye!

Use of Methods, Parameters, Modularity, Design, etc.

Part of your grade on this and ALL future programming projects in this course will be determined by how well you use multiple functions and parameter passing appropriately and how well you design a modular and functionally cohesive

program using the principles discussed in class. Large grade point penalties can be incurred for not setting up a modular, well designed program structure. This emphasizes good program structure, design, and fundamental software engineering principles.

Miscellaneous

Be sure to read and understand the sections in the Course Syllabus handout on general project requirements. Also be sure to study the style, documentation and formatting guidelines discussed in the *Programming Style Guidelines* handout and in the lecture.

What File To Turn In and How to Turn In Your Work using Blackboard

You must turn in your Java program source file in a zip file. The zip file must include an Employee.java file, JuniorEmployee.java, SeniorEmployee.java, Manager.java, EmployeeCompensation.java, and a MainProgram.java file which includes the main() method. The zip file's name must follow this format:

hw10_*yourLastNameLowerCase_yourFirstNameLowerCase*.zip

Hence, file names will look something like "hw10_tian_guanyu.zip"