CMIS 242 - Project 1

1. Specification

The first programming project involves writing a program that computes the salaries for a collection of employees of different types. This program consists of four classes.

1. The first class is **Employee**, which contains the employee's name and monthly salary which is specified in whole dollars. It should have three methods:

a. A constructor that allows the name and monthly salary to be initialized.
b. A method named **annualSalary** that returns the salary for a whole year.
c. A **toString** method that returns a string containing the name and monthly salary, appropriately labeled.

The **Employee** class has two subclasses: **Salesman** and **Executive**.

2. The **Salesman** class has an additional instance variable that contains the number of sold items. It should have the same three methods:

a. A constructor that allows the name, monthly salary and number of sold items to be initialized.
b. An overridden method **annualSalary** that returns the salary for a whole year. The salary for a salesman consists of the base salary computed from the monthly salary plus a commission. The commission is zero if the number of sold items is less than 200, equal to one month salary if the number of sold items is between 200 and 300 and equal to two month salary if the number of sold items is 300 or more.
c. An overridden **toString** method that returns a string containing the name, monthly salary and number of sold items, appropriately labeled.

3. The **Executive** class has an additional instance variable that reflects the current stock price. It should have the same three methods:

a. A constructor that allows the name, monthly salary and stock price to be initialized.
b. An overridden method **annualSalary** that returns the salary for a whole year. The salary for an executive consists of the base salary computed from the monthly salary plus a bonus. The bonus is $20,000 if the current stock price is greater than $100 and nothing otherwise.
c. An overridden **toString** method that returns a string containing the name, monthly salary and stock price, appropriately labeled.

4. Finally there should be a fourth class **P1Driver** that contains the **main** method. It should read in employee information from a text file **inputData.txt**. The file will contain employee information for only two years: 2015 and 2016. Each line of the text file will represent the information for one employee for one year. An example of how the text file will look is shown below:

```
2015 Employee Smithson,John 2000
2016 Salesman Jokey,Will 3000 236
2015 Executive Obama,Barack 5000 150
```

The year is the first data element on the line. Next is the type of the employee followed by the employee name and the monthly salary. For salesmen, the final value is the number of sold items and for executives the stock price. As the employees are read in, **Employee** objects of the appropriate type should be created and stored in an array depending upon the year. There should be two arrays, one corresponding to 2015 and one corresponding to year 2016. You may assume that the file will contain no more than 200 employee records for each year and that

the data in the file will be formatted correctly.

Once all the employee data is read in, a report should be displayed on the console for each of the two years. Each line of the report should contain all original data supplied for each employee together with that employee's annual salary for the year. The last line of the report should display the total number of employees and the average of all salaries for that year.

Your program should compile without errors.

The Google recommended Java style guide (https://google.github.io/styleguide/javaguide.html), should be used to format and document your code. Specifically, the following style guide attributes should be addressed:

- Header comments include filename, author, date and brief purpose of the program.
- In-line comments used to describe major functionality of the code.
- Meaningful variable names and prompts applied.
- Class names are written in UpperCamelCase.
- Variable names are written in lowerCamelCase.
- Constant names are in written in All Capitals.
- Braces use K&R style**.**

In addition the following design constraints should be followed:

- Declare all instance variables private
- Avoid the duplication of code

Test cases should be supplied in the form of a table with columns indicating what aspect is tested, the input values, expected output, actual output and if the test case passed or failed. This table should contain 5 columns with appropriate labels and a row for each test case. Note that the actual output should be the actual results you receive when running your program and applying the input for the test record. Be sure to select enough different kinds of employees and situations to completely test the program.

2. Submission Requirements

Submit the following to the Project 1 assignment area no later than the due date listed in your LEO classroom.

1. All **.java** source files (**no other file types should be submitted**) and the **inputData.tx**t file. The source code should use Java code conventions and appropriate code layout (white space management and indents) and comments. All submitted files may be included in a .zip file.
The input file will be generated by the students using a simple text editor such as Notepad.

2. The solution description document **P1SolutionDescription** (.pdf or .doc / .docx) containing the following:
(1) Assumptions, main design decisions, error handling;
(2) Test cases table
(3) Screen captures showing successful program compilation and test cases execution. Each screen capture should be properly labeled, clearly indicated what the screen capture represents.
(4) Lessons learned from the project;

3. Grading Rubric

The following grading rubric will be used to determine your grade:

Attribute	Meets	Does not meet
Employee class	**25 points** a) Contains the employee's name and monthly salary, which is specified in whole dollars. b) Contains a constructor that allows the name and monthly salary to be initialized. c) Contains a method named annualSalary that returns the salary for a whole year. d) Contains a toString method that returns a string containing the name and monthly salary, appropriately labeled.	**0 points** a) Does not contain the employee's name and monthly salary, which is specified in whole dollars. b) Does not contain a constructor that allows the name and monthly salary to be initialized. c) Does not contain a method named annualSalary that returns the salary for a whole year. d) Does not contain a toString method that returns a string containing the name and monthly salary, appropriately labeled.
Salesman class	**15 points** a) Is a subclass of Employee. b) Contains an additional instance variable that contains the number of sold items. c) Contains a constructor that allows the name, monthly salary and annual sales to be initialized. d) Contains an overridden method annualSalary that returns the salary for a whole year as indicated in the project specification. e) Contains an overridden toString method that returns a string containing the name, monthly salary and annual sales, appropriately labeled.	**0 points** a) Is not a subclass of Employee. b) Does not contain an additional instance variable that contains the number of sold items. c) Does not contain a constructor that allows the name, monthly salary and annual sales to be initialized. d) Does not contain an overridden method annualSalary that returns the salary for a whole year as indicated in the project specification. e) Does not contain an overridden toString method that returns a string containing the name, monthly salary and annual sales, appropriately labeled.

Executive class	**15 points** a) Is a subclass or Employee. b) Contains an additional instance variable that reflects the current stock price. c) Contains a constructor that allows the name, monthly salary and stock price to be initialized. d) Contains an overridden method annualSalary that returns the salary for a whole year as shown in the project specification. e) Contains an overridden toString method that returns a string containing the name, monthly salary and stock price, appropriately labeled.	**0 points** a) Is not a subclass or Employee. b) Does not contain an additional instance variable that reflects the current stock price. c) Does not contain a constructor that allows the name, monthly salary and stock price to be initialized. d) Does not contain an overridden method annualSalary that returns the salary for a whole year as shown in the project specification. e) Does not contain an overridden toString method that returns a string containing the name, monthly salary and stock price, appropriately labeled.
P1Driver class	**25 points** a) Contains the main method. b) Reads in employee information from a text file. c) As the employees are read in, Employee objects of the appropriate type are created and stored in one of two arrays depending upon the year. d) Once all the employee data is read in, a report is displayed on the console for each of the two years. e) Each line of the report contains all original data supplied for each employee together with the employee's annual salary for the year. f) For each of the two years, the number of employees and the average of all salaries for all employees for that year is	**0 points** a) Does not contain the main method. b) Does not reads in employee information from a text file. c) As the employees are read in, Employee objects of the appropriate type are not created and stored in one of two arrays depending upon the year. d) Once all the employee data is read in, a report is not displayed on the console for each of the two years. e) Each line of the report does not contain all original data supplied for each employee together with the employee's annual salary for the year. f) For each of the two years, the number of employees and the

	computed and displayed.	average of all salaries for all employees for that year is not computed and displayed.
Test Cases	**10 points** a) Test cases are supplied in the form of table with columns indicating test case objective, the input values, expected output, actual output and if the test case passed or failed. b) Enough employees selected to completely test the program. c) Enough test cases considered to completely test the program d) Test cases table is included in the supporting word or PDF documentation.	**0 points** a) No test cases were provided.
Documentation and Style guide	**10 points** a) Solution description document **P1SolutionDescription** includes all the required sections appropriate titled. b) Solution description document **P1SolutionDescription** includes project specific, meaningful information. **Source code** criteria c) Header comments include filename, author, date and brief purpose of the program. d) In-line comments used to describe major functionality of the code. e) Meaningful variable names and prompts applied. f) Class names are written in UpperCamelCase.	**0 points** a) No **solution description document** is included. **Source code** criteria b) Java style guide was not used to prepare the Java code. c) All instance variables not declared private. d) Duplication of code was not avoided.

	g) Variable names are written in lowerCamelCase. h) Constant names are in written in All Capitals. i) Braces use K&R style. i) Declare all instance variables private. k) Avoids the duplication of code.	