Project 1

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, employee's age and monthly salary which is specified in whole dollars. It should have three methods:

    a. A constructor that allows the name, age 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, the age 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, age, 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 75, equal to one month salary if the number of sold items is between 75 and 150 and equal to two month salary if the number of sold items is 151 or more.
    c. An overridden **toString** method that returns a string containing the name, the age, 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, the age, 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 $45,000 if the current stock price is greater than $250 and nothing otherwise.
    c. An overridden **toString** method that returns a string containing the name, the age, 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: 2016 and 2017. 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:

```
2016 Employee Smithson,John 54 2000
2017 Salesman Jokey,Will 42 3000 136
2017 Executive Trump,Donald 72 5000 350
```

The year is the first data element on the line. Next is the type of the employee followed by the employee name, followed by age 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 2016 and one corresponding to year 2017. You may assume that the file will contain no more than 500 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 lines of the report should display for each year, the total number of employees, the average of all salaries and the name of the person with the largest salary.

Additional instance variables and/or methods could be defined if necessary (give reasons in the solution description document).

Note: All code should compile and run without issue.

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 five columns indicating what aspect is tested, the input values, the expected output, the actual output and if the test case passed or failed. This table should contain five 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.

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 input file will be generated by the students using a simple text editor such as Notepad.
The source code should use the recommended Java style and code conventions and appropriate code layout (white space management and indents) and comments.
All submitted files may be included in a .zip file.

2. The solution description document **P1SolutionDescription** (.pdf or .doc / .docx) containing the following sections:
(1) Assumptions, main design decisions and 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;

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, age and monthly salary, which is specified in whole dollars. b) Contains a constructor that allows the name, age 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, the age and monthly salary, appropriately labeled.	**0 points** a) Does not contain the employee's name, age and monthly salary, which is specified in whole dollars. b) Does not contain a constructor that allows the name, age 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, the age and monthly salary, appropriately labeled. e) Code does not compile
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, the age, 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, the age, 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, the age, 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, the age,

		monthly salary and annual sales, appropriately labeled. f) Code does not compile
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, the age, monthly salary and stock price 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, the age, 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, the age, 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 indicated in the project specification. e) Does not contain an overridden toString method that returns a string containing the name, the age, monthly salary and stock price, appropriately labeled. f) Code does not compile
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 displays 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 that employee's annual salary for the year. f) For each of the two years, the number	**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 that employee's annual salary for the

	of employees, the average of all salaries and the name of the person with the largest salary are computed and displayed.	year.
		f) For each of the two years, the number of employees, the average of all salaries and the name of the person with the largest salary are not computed and displayed.
		g) Code does not compile
Test Cases	**10 points**	**0 points**
	a) Test cases are supplied in the form of table with five columns indicating test case objective, the input values, expected output, actual output and if the test case passed or failed.	a) No test cases were provided.
	b) Enough employees selected to completely test the program.	
	c) Enough test cases considered to completely test the program	
	d) Test cases were included in the supporting word or PDF documentation.	
Documentation and Style guide	**10 points**	**0 points**
	a) Solution description document **P1SolutionDescription** includes all the required sections appropriate titled and gives relevant and meaningful details in each section.	a) No **solution description document** is included.
	Source code criteria	**Source code** criteria
	b) Header comments include filename, author, date and brief purpose of the program.	b) No header comments including filename, author, date and brief purpose of the program
	c) In-line comments used to describe major functionality of the code.	c) No in-line comments used to describe the major functionality of the code.
	d) The recommended Java style is used to prepare the Java code.	d) The recommended Java style is not used to prepare the Java code.
	e) Meaningful variable names and prompts applied.	e) No meaningful variable names and prompts.
	f) Declare all instance variables private.	f) Not all instance variables are declared as private
	g) Avoids the duplication of code.	g) Source files contain duplicated code