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 the Employee class, 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.

2. The Employee class has two subclasses. The first is Salesman. It has an additional instance variable that contains the annual sales in whole dollars for that salesman. It should have the same three methods:
    a. A constructor that allows the name, monthly salary and annual sales 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 computed as 2% of that salesman's annual sales. The maximum commission a salesman can earn is $20,000.
    c. An overridden toString method that returns a string containing the name, monthly salary and annual sales, appropriately labeled.

3. The second subclass is Executive. It 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 $30,000 if the current stock price is greater than $50 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 that contains the main method. It should read in employee information from a text file. 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:

    2014 Employee Smith,John 2000
    2015 Salesman Jones,Bill 3000 100000
    2014 Executive Bush,George 5000 55

The year is the first data element on the line. The file will contain employee information for only two years: 2014 and 2015. Next is the type of the employee followed by the employee name and the monthly salary. For salesmen, the final value is their annual sales and for executives the stock price. As the employees are read in, Employee objects of the appropriate type should be created and they should be stored in one of two arrays depending upon the year. You may assume that the file will contain no more than ten 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. For each of the two years, an average of all salaries for all employees for that year should be computed and displayed.

The google recommended Java style guide, provided as link in the week 2 content, 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 table with columns indicating the input values, expected output, actual output and if the test case passed or failed. This table should contain 4 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 to completely test the program.

Note: All code should compile and run without issue.

Submission requirements
Deliverables include all Java files (.java) and a single word (or PDF) document. The Java files should be named appropriately for your applications. The word (or PDF) document should include screen captures showing the successful compiling and running of each of the test cases. Each screen capture should be properly labeled clearly indicated what the screen capture represents. The test cases table should be included in your word or PDF document and properly labeled as well.

Submit your files to the Project 1 assignment area no later than the due date listed in your LEO classroom. You should include your name and P1 in your word (or PDF) file submitted (e.g. firstnamelastnameP1.docx or firstnamelastnameP1.pdf)

Grading Rubric:

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

Attribute	Meets	Does not meet
Employee Class	25 points	0 points

	Contains the employee's name and monthly salary, which is specified in whole dollars. Contains a constructor that allows the name and monthly salary to be initialized. Contains a method named annualSalary that returns the salary for a whole year. Contains a toString method that returns a string containing the name and monthly salary, appropriately labeled.	Does not contain the employee's name and monthly salary, which is specified in whole dollars. Does not contain a constructor that allows the name and monthly salary to be initialized. Does not contain a method named annualSalary that returns the salary for a whole year. Does not contain a toString method that returns a string containing the name and monthly salary, appropriately labeled. Code does not Compile.
Salesman class	15 points Is a subclass of Employee. Contains an additional instance variable that contains the annual sales in whole dollars for that salesman. Contains a constructor that allows the name, monthly salary and annual sales to be initialized. Contains 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 computed as 2% of that salesman's annual sales. The maximum commission a salesman can earn is $20,000.	0 points Is not a subclass of Employee. Does not contain an additional instance variable that contains the annual sales in whole dollars for that salesman. Does not contain a constructor that allows the name, monthly salary and annual sales to be initialized. Does not contain 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 computed as 2% of that salesman's annual sales. The maximum commission a salesman can earn is $20,000.

	Contains an overridden toString method that returns a string containing the name, monthly salary and annual sales, appropriately labeled.	Does not contain an overridden toString method that returns a string containing the name, monthly salary and annual sales, appropriately labeled. Code does not Compile.
Executive Class	15 points Is a subclass or Employee. Contains an additional instance variable that reflects the current stock price. Contains a constructor that allows the name, monthly salary and stock price to be initialized. Contains 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 $30,000 if the current stock price is greater than $50 and nothing otherwise. Contains an overridden toString method that returns a string containing the name, monthly salary and stock price, appropriately labeled.	0 points Is not a subclass or Employee. Does not contain an additional instance variable that reflects the current stock price. Does not contain a constructor that allows the name, monthly salary and stock price to be initialized. Does not contain 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 $30,000 if the current stock price is greater than $50 and nothing otherwise. Does not contain an overridden toString method that returns a string containing the name, monthly salary and stock price, appropriately labeled. Code does not Compile.
Main Method Class	25 points Contains the main method. Reads in employee information from a text file. As the employees are read in, Employee objects of the	0 points Does not contain the main method. Does not reads in employee information from a text file.

	appropriate type are created and stored in one of two arrays depending upon the year. Once all the employee data is read in, a report displays on the console for each of the two years. Each line of the report contains all original data supplied for each employee together with that employee's annual salary for the year. For each of the two years, an average of all salaries for all employees for that year is computed and displayed.	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. Once all the employee data is read in, a report is not displayed on the console for each of the two years. Each line of the report does not contain all original data supplied for each employee together with that employee's annual salary for the year. For each of the two years, an average of all salaries for all employees for that year is not computed and displayed. Code does not Compile.
Test Cases	10 points Test cases are supplied in the form of table with columns indicating the input values, expected output, actual output and if the test case passed or failed. Enough employees selected to completely test the program. Test cases were included in the supporting word or PDF documentation.	0 points No test cases were provided.
Documentation and Style guide	10 points Screen captures were provided and labeled for compiling your code, and running each of your test cases.	0 points No documentation included. Java style guide was not used to prepare the Java code.

	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**.** Declare all instance variables private. Avoids the duplication of code.	All instance variables not declared private. Duplication of code was not avoided.