Programming assignment 2

Employee Management System 1.0

Objective:

To successfully be able to apply the programming concepts, you have learned from previous lectures and labs in this class up to this point, in writing a code (program) that runs with no errors and gives the correct output.

Skills needed:

- Knowledge of programming concepts from previous lectures and labs in this class up to this point. Especially creating a class and methods.
- Hands on Eclipse.

Task:

You will need to develop a system that can track employee information for two Organizations (Google and Microsoft).

General view of what your system should be able to do:

- Add employees to the system, the employee must be working for one of the companies in the system (Google and Microsoft).
- Update employee's information.
- Print employee's information.
- Print the number of all employees of a specific company.
- Print the number of all employees in the system (in both companies).

System Structure:

Employee Class:

The Employee information you must track is as follows:

- ID (This is generated randomly)
- Name
- Gender
- Job Title
- Organization they work for
- Birthday

As for the Organization that the Employee works for, you must also track this information:

- Organization Name
- Number of Employees of each organization

Note: If an employee is added to a specific company, only the number of employees in that company is incremented by one, not the other.

Caller class:

The caller class must create instances of the Employee class and be able to comply to the general view of the system above (adding employees, and displaying employee information, etc.).

Test Run

```
~~~~ Welcome to the Employee Management System 1.0 ~~~~
Please enter #1 employee's name: John Wick
Please enter #1 employee's gender: male
Please enter #1 employee's job title: Software Engineer
Please choose an organization for employee #1 as a number (1: Google, 2: Microsoft): 1
Please enter #1 employee's birthday (Formet MM/DD/YYYY): 12/13/1234
Thank you... :)
------------------------------
Please enter #2 employee's name: Katniss Everdeen
Please enter #2 employee's gender: female
Please enter #2 employee's job title: Android Developer
Please choose an organization for employee #2 as a number (1: Google, 2: Microsoft): 4
The organization you entered is not in the system, please choose either 1 or 2: 1
Please enter #2 employee's birthday (Formet MM/DD/YYYY): 122/23/54
Thank you... :)
------------------------------
Please enter #3 employee's name: Jack Sparrow
Please enter #3 employee's gender: male
Please enter #3 employee's job title: Pirate
Please choose an organization for employee #3 as a number (1: Google, 2: Microsoft): 2
Please enter #3 employee's birthday (Formet MM/DD/YYYY): 123/27/1232
Thank you... :)
------------------------------

################## Stats ##################
Total number of employees in the system is: 3
Microsoft: 1                    Google: 2
.:.:.:.:. Priniting all employees in the system .:.:.:.:.
EID: -317183082
Name: John Wick
Gender: male
Job Title: Software Engineer
Organization: Google
Birthday: 12/13/1234
---------
EID: -1940766990
Name: Katniss Everdeen
Gender: female
Job Title: Android Developer
Organization: Google
Birthday: 122/23/54
---------
EID: 1056230567
Name: Jack Sparrow
Gender: male
Job Title: Pirate
Organization: Microsoft
```

Your programs and all further programs should have **program description header** that goes at the top of the file which gives information about the programmer and program.

```
////
// Name: Bob Programmer
// Section: A, B, or S
// Program Name: Hello World
//
// Description: A brief description of the program.  What does the
//  program do (not how it does it: for example, it uses loops)?  Does
//  the program get input?  What kind?  What information is output
//  from the program and to where (screen or file)
////

import…
```

Run your program with different inputs. Take a screenshot of your program's output to the bottom of your code. This will be needed in your report. In order to know how to write up your report, take a look at the attached word document and follow the instructions.

Grading Criteria:

I will be looking at the following things: **(Total Points: 100)**

- Do you have a program description header (in your code) as shown in the above example? **(5 pts)**
- Did you get the correct output? **(50 pts)**
- Did you choose good variable names? **(20 pts)**
- Did you indent the code properly? **(5 pts)**
- Is your output easy to read? **(5 pts)**
- Did you attach the report with the output from the test runs in your report? **(15 pts)**