Programming Assignment 1

http://www.hibbsandassociates.com/tiny_mce/images/employees.png

Objectives:

- Practice implementing classes
- Practice creating and using objects
- Practicing using classes
- Practice creating tester programs

Assignment:

For this assignment, you are going to simulate a company roster. You will need 3 separate files: one for the Employee class, one for the EmployeeTester class, and one for the Company class.

Employee Class

This class will simulate employees.

You will need 4 instance variables for:

- Employee's name
- Employee's ID number
- Department employee works in
- Employee's position in the company

You will need a getter and setter to access each of the instance variables (8 methods total).

You will need to include an argument constructor and a no-argument constructor to initialize the instance variables.

EmployeeTester Class

This class will test the methods and constructors of the Employee Class.

Create 2 objects, one with the argument constructor and one with the no-argument constructor.

On the object constructed with the argument constructor, call all the getter methods to make sure the instance variables were initialized correctly.

On the object constructed with the no-argument constructor, call all the setter methods and set the instance variables to the values of your choosing. Then, call all the getter methods to make sure the setters worked correctly.

The output that should be produced by this class is shown below in the sample execution section.

Company Class

This class will use the Employee class to simulate 3 employees. Below is the employee information you will use:

Name	ID Number	Department	Position
Susan Meyers	47899	Accounting	Vice President
Mark Jones	39119	IT	Programmer
Joy Rogers	81774	Manufacturing	Engineer

Create 2 objects of the Employee class using the argument constructor and 1 object with the no-argument constructor. Set the values of the instance variables to the information in the table.

Using the getter methods, display each employee's information in a table, shown below.

The output that should be produced by this class is shown below in the sample execution section.

Sample Execution:

EmployeeTester Class Output:

```
TESTING ARGUMENT CONSTRUCTOR
----------------------------

Tyler Greer
Expected: Tyler Greer

12345
Expected: 12345

Computer Science
Expected: Computer Science

Instructor
Expected: Instructor

TESTING NO-ARGUMENT CONSTRUCTOR
-------------------------------

Desiree Wilson
Expected: Desiree Wilson

45678
Expected: 45678

Orthopaedics
Expected: Orthopaedics

Registered Nurse
Expected: Registered Nurse
```

Company Class Output:

```
Name                ID Number           Department          Position
-------------------------------------------------------------------------------
Susan Meyers        47899               Accounting          Vice President
Mark Jones          39119               IT                  Programmer
Joy Rogers          81774               Manufacturing       Engineer
```

Requirements:

- Use an updated comment block
- Your program should use the following comment block at the very beginning of your program.

// Name: Your Name **Date Assigned: Fill in**
//
// Course: CSCI 2003 42733 **Date Due: Fill in**
//
// Instructor: Ms. Greer
//
// File name: Fill in
//
// Program Description: Brief description of what the program does.

- Use appropriate comments throughout the program
- Make good use of whitespace
- Your output should look exactly like the sample output if using the same data.

Deliverables:

- Employee.java file
- EmployeeTester.java file
- Company.java file
- Upload all 3 files to Moodle

Grading:

Total Points	10 points
Employee Class	**4 points**
Declare instance variables correctly	1 point
Constructors are correct	1 point
Getters are correct	1 point
Setters are correct	1 point
EmployeeTester Class	**3 points**
Tests both types of constructors	1 point
Tests all getters correctly	1 point
Tests all setters correctly	1 point
Company Class	**3 points**
Creates 3 objects	1 point
Initializes instance variables correctly	1 point
Uses getters to create a table of the employees' information	1 point
Not enough comments/whitespace	**-1 point**
Output does not match the sample executions given in the assignment	**-1 point**
Bad variable names	**-1 point**