CS101- Algorithms and Programming I

Lab 08

Lab Objectives: Classes and Objects; Object vs. reference, comparing objects, copying objects.

- ❑ For all labs in CS 101, your solutions must conform to the CS101 style guidelines.
- ❑ Remember to include javadoc comments for each class and method.

---

a. Create a new project Lab08. Download class `Project` and add to Lab08 project. You should not make any changes to this class.

b. Create a class `Department`:
   - **Instance Data Members:**
     - `deptName`: stores the name of the department.
     - `deptCode`: stores the code of the department.
   - **Methods:**
     - **Constructor:**
       - Initializes the department name and department code using the ones passed as parameters.
     - **Accessor / Mutator methods for:**
       - `deptName, deptCode.`
     - **Other methods:**
       - `equals`(): instance method that takes an Object as a parameter, and returns true if the target department and the Department passed as a parameter are the same, false if not.
       - `toString`(): returns a String representation of a department. See sample output for format details.

c. Create a class `DepartmentTest`.

   Write a main method to create and compare Department instances using == and .equals.

   - Create two references to a single Department instance,
   - Create two references to two individual instances with different properties,
   - Create two references to two individual objects with identical properties.

   You should run your code without `.equals` implementation in your `Department` class (comment your `.equals` method in `Department` class). Do you get the same results with == and .equals? Can you explain why?

   Now write `.equals` method in your `Department` class. Two department instances are considered the same if they have the same department name and the same department code. Run your comparison tests again and explain whether there are any changes in the output.

,

d. Create a class `Employee`:
    - Constant data member:
        - WORKING_DAYS: there are 261 working days per year.

    - **Instance Data Members:**
        - `employeeName`: stores the name of the employee.
        - `dailyRate`: stores the double daily pay rate of the employee.
        - `department`: stores the Department of the employee.
        - `project`: stores the project the employee has been assigned to.

    - **Methods:**
        - **Constructor:**
            - Initializes the employee name, rate and project using the ones passed as parameters. Also takes the department name and code as parameters, initializes a new department using the ones passed as a parameter.
        - **Constructor:**
            - Copy constructor: creates a new `Employee` object using the data from the `Employee` passed as a parameter. The new employee will be assigned to the same project.

        - **Accessor / Mutator methods for:**
            - `employeeName, dailyRate, department, project.`
        - **Other methods:**
            - `calculateYearlySalary`(): Calculates and returns yearly salary.
            - `toString`(): returns a String representation of an Employee. See sample output for format details.

e. Create and application, EmployeeApp that does the following:
        - Create a Project.
        - Create 3 Employees who are assigned to the project.
        - Create a new Employee that is a copy of the first.
        - Display the 4 Employees.
        - Compare the Department of all Employees, and display Employees with matching Departments.

Sample Output:

_____

```
Employees:

Employee Name: Karakus, Zana Yearly Salary: 56115.0
DeptName: Information Technology Dept Code: IT
Project Name: SunMarkets POS Implementation
Project ID: 2021-1000
Project Type: a
Team Size: 1
Estimated Project Cost: 176436.0


Employee Name: Rocca, Denis Yearly Salary: 45675.0
DeptName: Human Resources Dept Code: HR
Project Name: SunMarkets POS Implementation
Project ID: 2021-1000
Project Type: a
Team Size: 1
Estimated Project Cost: 176436.0


Employee Name: Anders, Jamie Yearly Salary: 71775.0
DeptName: Human Resources Dept Code: HR
Project Name: SunMarkets POS Implementation
Project ID: 2021-1000
Project Type: a
Team Size: 1
Estimated Project Cost: 176436.0


Employee Name: Karakus, Zana Yearly Salary: 56115.0
DeptName: Information Technology Dept Code: IT
Project Name: SunMarkets POS Implementation
Project ID: 2021-1000
Project Type: a
Team Size: 1
Estimated Project Cost: 176436.0

--------- end employee list ----------

_____________________
Employees with Matching Departments (1)

Employee Name: Karakus, Zana Yearly Salary: 56115.0
DeptName: Information Technology Dept Code: IT
Project Name: SunMarkets POS Implementation
Project ID: 2021-1000
Project Type: a
Team Size: 1
Estimated Project Cost: 176436.0
```

Employee Name: Karakus, Zana Yearly Salary: 56115.0
DeptName: Information Technology Dept Code: IT
Project Name: SunMarkets POS Implementation
Project ID: 2021-1000
Project Type: a
Team Size: 1
Estimated Project Cost: 176436.0

_____
Employees with Matching Departments (2)

Employee Name: Rocca, Denis Yearly Salary: 45675.0
DeptName: Human Resources Dept Code: HR
Project Name: SunMarkets POS Implementation
Project ID: 2021-1000
Project Type: a
Team Size: 1
Estimated Project Cost: 176436.0

Employee Name: Anders, Jamie Yearly Salary: 71775.0
DeptName: Human Resources Dept Code: HR
Project Name: SunMarkets POS Implementation
Project ID: 2021-1000
Project Type: a
Team Size: 1
Estimated Project Cost: 176436.0