C S 272/463 Introduction to data structures
Fall 2019
Lab 2: Class definition and usage (OOP)

1 Learning objectives

Objective 1 (class), Objective 5, Objective 6, Objective 7

2 Requirements

Specify and design a class called `Employee` and implement it in a file `Employee.java`.

1. (12pts) Implement the class to contain the following instance variables:

   (1) the employee name (data type: `String`)
   (2) the employee no (data type: `int`)
   (3) the employee age (data type: `int`)
   (4) the employee state (data type: `String`)
   (5) the employee zip code (data type: `int`)
   (6) the advisors (data type: array of `Integer`) to keep the employee nos of this employee's advisors, where each employee can have at most 3 advisors.

2. (88pts) You are required to implement the following methods:

   (1) (4pts) One no-argument constructor.
   This constructor sets null to employee name, state, and zip codes, sets 0 to employee number and employee age, and sets advisor array to be null.

   ```
   public Employee()
   ```

   (2) (10pts) One copy constructor that uses the given parameter `obj` to set the current object's instance variables. Please be very careful with `String` copy. The precondition is that `obj` should not be null and should be an instance of `Employee`.

   ```
   public Employee (Object obj)
   ```

   (3) (24pts) The get and set methods of all the instance variables.

   (4) (10pts) `toString()` method to generate a string representation of an employee.

   ```
   public String toString()
   ```

   This method should organize the String information in the order of
   employee name, employee no, age, state, zip code, and list of advisors' employee nos.

   (5) (12pts) `equals` method
   This method returns true if the given object's employee no is the same as the no of the given employee instance which activates this method.
   Otherwise, this method returns false.
   The precondition is that `obj` should not be null and should be an instance of the `Employee` class.

   ```
   public boolean equals(Object obj)
   ```

   (6) (15pts) A static method `getAllAdvisors` to get all the DISTINCT advisors of two employees which are the input parameters.
   The preconditions are that neither `e1` nor `e2` should be null.

   ```
   public static int[] getAllAdvisors(Employee e1, Employee e2)
   ```

(7) (13pts) `main()` method in `Employee.java` to thoroughly test your code.
Design test cases, put them in your main method, run your program through the test cases.

```
public static void main(String[] args)
```

3 Note

- **Specifications** for all your classes and methods:
  Please properly explain (1) the functionality of the methods, (2) the parameters, (3) the return values, (4) the pre-conditions if there is any;
  Please use inline comments, meaningful variable names, indentation, formatting, and whitespace throughout your program to improve its readability.
- You can (but are not required to) design and implement other facilitating methods (E.g., other get and set methods, toString method) to finish the implementation of the required methods.

4 Submission

Submit through canvas a zipped file containing your java file(s) (not `.class` files).

5 Grading Criteria

(1) The score allocation is beside the questions.

(2) Please make sure that you test your code **thoroughly** by considering all possible test cases.
Your code may be tested using more test cases.