C S 272/463 Introduction to Data Structures

Lab 2: Class Definition and Usage (OOP)

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

1. (12 points) Implement the class that contains 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. You are required to implement the following methods:
    (1) (4 pts) One no-argument constructor.
    This constructor sets null to all the variables with non-primitive data types and sets zero to variables with int.
    ```
    public Employee()
    ```

    (2) (10 pts) 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) (10) The clone method. Need to be deep copy.

    (4) (14 pts) The get and set methods of all the instance variables.

    (5) (10 pts) 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.

    (6) (12 pts) 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)
    ```

    (7) (15 pts) 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)
    ```

(8) (13 pts) 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, 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 instructions**
Submit through canvas a zip file consisting of 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.