Programming and Data Structures
Active Learning Activity 1: Review of Java Fundamentals and OOP concepts
Due date: February 12, 2021

Activity Objectives

At the end of this activity, students should be able to:

1. Create classes in Java with the specified data members and methods
2. Create subclasses from a super class using inheritance
3. Override methods from the super class in the sub classes
4. Use super constructor in the subclasses
5. Use super and subclasses in a test program
6. Define methods that accept an array of objects of the super type and the array holds instances of the super and subclasses

Activity

1. Define a class named ***Person*** that contains four instance variables of type **String** to store the name, address, phone number, and email of a person. Create two constructors for the class: a default constructor that initializes **name**, **address**, **phone**, and **email** to the string **"none"**, and a constructor with four parameters of type **String** to initialize the variables **name**, **address**, **phone**, and **email**. Define getters and setters for all the instance variables of the class and a method named ***toString*** that returns the person information in a formatted string as shown below.

```
Name: Helen Brown
Address: 222 10th Street, Bethlehem
Phone: (610)334-2288
Email: hbrown@gmail.com
```

2. Define a class ***Student*** that is derived from the class ***Person*** and has three additional instance variables for the ***id***, ***major***, and ***gpa*** of the student. Implement two constructors, accessor and mutator methods for the class. Override the ***toString()*** method to return the student information in a formatted string as shown below.

```
Name: Gary Leister
Address: 972 4th Street, Emmaus
Phone: (202)331-7177
Email: gleister@gmail.com
ID: 12345
```

```
Major: CSE
GPA: 3.50
```

3. Define a class ***Employee*** that extends the class ***Person*** and has three additional instance variables for the **id**, **title**, and **annual salary** of the employee. Define two constructors, accessor and mutator methods. Override **toString()** method to return the employee information in a formatted string as shown below.

```
Name: Beth Down
Address: 234 Main Street, Philadelphia
Phone: (610)222-4433
Email: bdown@gmail.com
ID: 33442
Title: Systems Administrator
Annual Salary: $75000.00
```

4. Define a class ***Faculty*** that extends class ***Employee*** and has one additional instance variable for the **rank** of the faculty. Define two constructors, accessor and mutator methods. Override the **toString()** method to return the faculty information in a formatted string as shown below.

```
Name: Mark Jones
Address: 21 Orchid Street, Bethlehem
Phone: (610)333-2211
Email: mjones@gmail.com
ID: 22222
Title: Faculty
Annual Salary: $90000.00
Rank: Professor
```

5. Finally, create a main class named **Test**. In the main method, create an array **objects** of type **Person** and size 4. Create four objects of type ***Person***, ***Student***, **Employee**, and **Faculty**, and store them in the array **objects**. Use the sample information provided above to create each object. Then print the information of the four objects by calling the method **printArray**. Sort the four objects based on the names by calling the method **selectionSort** and call **printArray** again to display the sorted objects.

6. In the class **`Test`**, define the method **`printArray`** that accepts an array of type **`Person`** and prints the information of the objects in the array. The method should print the return value of **`toString()`** method invoked on each element of the array.
7. In the class **`Test`**, define the method **`selectionSort`** that accepts an array of type **`Person`** and sorts the objects in the array based on the data member **`name`**. Use the selection sort algorithm for the method.

In all the derived classes, the constructors should call the corresponding constructors from the base class and the method **`toString()`** should call **`toString()`** methods from the base classes. Your code must include Javadoc comments for each class and each method.

Submit the files **`Person.java`**, **`Student.java`**, **`Employee.java`**, **`Faculty.java`,** and **`Test.java`**.