Hands-On MidTerm TEST-B

Be sure to read the following general instructions carefully:

- This lab test must be completed individually by all the students.
- Save your program periodically just in case that your PC crashes.
- All steps are required for the test
- The midterm carries 25% marks towards final grade

Exercise

Start a new Eclipse Java project. Use default package.

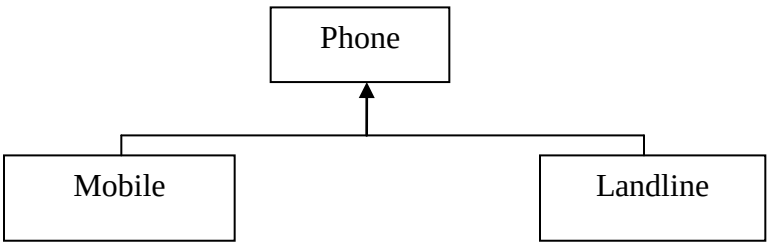
1. Create a public Java class *MidtermTest* with *main* method. This will be your driver class. You will create all other classes as non-public and create in the same Java file.

2. Create an abstract class called *Phone*
   The class should declare the following variables:
   - an instance variable that describes the *phoneNumber* - String
   - an instance variable that describes the *serialNumber* - String
   - an instance variable that describes the *brand* - String
   - an instance variable that describes the *billingRate* - double
   - an instance variable that describes the *minutes – integer*

   Provide a *toString()* method that returns the information stored in the above variables.

   Create the **getter** and **setter** methods for each instance variable except *billingRate*. Provide the necessary constructors. Include *an **abstract** method **setBillingRate(double rate)*** to determine the rate for a phone.

3. Create two subclasses called *Mobile* and *Landline*.



These subclasses should override the abstract methods *setBillingRate* of class *Phone*.

Use the following rule for setting the rate for a phone:

- Mobile will have a 5% discount
- Landline will have a fixed rate (specified by user)

4. In driver class *MidtermTest*
    - Add a method *printInfo* to print data from a Phone array.
    - In main method create 2 objects of *Mobile* and 1 object of *Landline*, store references to an array and call method *printInfo* to print data. When you run your code, it should print information from the objects that you had created.

Example: Following is an example input/output of running driver program

Mobile added with

- phonenumber "416-111-2222", serial number "12345", brand "brand1" and minutes 10, billing rate 0.50
- phonenumber "647-111-3333", serial number "66789", brand "brand1" and minutes 20, billing rate 0.60

And Landline with

- phonenumber "416-222-3333", serial number "000001", brand "brand4" and minutes 40, billing rate 0.10

Will output:

```
phonenumber=416-111-2222 serialnumber=12345 brand=brand1 rate=0.475 minutes=10
phonenumber=647-111-3333 serialnumber=66789 brand=brand1 rate=0.57 minutes=20
phonenumber=416-222-3333 serialnumber=000001 brand=brand4 rate=0.1 minutes=40
```

Compile your code, test it and then upload the Java file *MidtermTest.java* to eCentennial before the end of the test.

Evaluation:

Functionality	
Correct implementation of classes (instance variable declarations, constructors, getter and setter methods, etc.)	30%
Correct implementation of Inheritance/Polymorphism	20%
Correct implementation of driver classes (declaring and creating objects, calling their methods, interacting with user, displaying results)	35%
Correct naming of variables, methods, classes, etc.	5%
Friendly input/output	10%
Total	100%