

Northeastern Illinois University
CS207, Object-Oriented Programming and Data Structures, Summer 2022
Homework 1

Due date: Thursday 6/9/2022 at 11:59 p.m.

Problem 1:

Complete the Clothing class, the class should have the following instance variables, constructor(s) and instance methods:

- An instance variable of type double named price.
- An instance variable of type String named type.
- An instance variable of type String named size.
- A default “no-arg” constructor that sets the price to 29.9, the type to “T-Shirt” and the size to “Medium”.
- An overloaded constructor that takes three parameters and set the instance variables accordingly.
- A method named increasePrice that takes no parameters and returns no value, the method increases the instance variable price by 5.
- An overloaded method increasePrice that takes a double parameter x and returns no value, the method increases the instance variable price by the value of parameter x.
- A method named details which takes no parameters, returns no value, the method prints the Clothing item details in the format shown in the sample output.
- **Your output should look exactly the same as follows:**

Item: T-Shirt
Size: Medium
Price: \$29.9

Item: T-Shirt
Size: Medium
Price: \$34.9

Item: T-Shirt
Size: Medium
Price: \$41.9

Item: Jeans
Size: Small
Price: \$31.2

Item: Jeans
Size: Small
Price: \$41.2

Item: Jeans
Size: Small
Price: \$45.2

Problem 2:

Point
- x: int - y: int
+ Point() + Point(xVal: int, yVal: int) + getX(): int + getY(): int + setpoint(newX: int, newY: int): void + toString(): String + distance(Point pnt): double

Given the UML diagram above, complete the Point class that models a point on a Cartesian plane, the class should have the following instance variables, constructor(s) and instance methods:

- Two private integer instance variables, x and y.
- A “no-arg” constructor that creates a point with the coordinates (0, 0).
- An overloaded constructor that takes two integer parameters xVal and yVal and set the instance variables accordingly.
- A getter method for each instance variable.
- A method named setPoint that takes two integer parameters newX and newY and set the instance variables accordingly.
- A method named toString which takes no parameters, and returns a string with the coordinates of the point within parentheses and comma separated, for example: (2, 5)
- A method named distance which takes one class parameter of type Point, the method returns the distance (a double value) from the (x, y)-

location of the Point object that calls the method to the (x, y)-location of the Point object passed in.

The distance between two point is calculated by the formula:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Hints:

Use `Math.pow(n, 2)` to raise a number to a power.

Use `Math.sqrt(number)` to find the square root of a number.

- **Your output should look exactly the same as follows:**

```
The distance from (0, 0) to (3, 4) is: 5.0
```

```
P1 new coordinates are:
```

```
x = 1
```

```
y = 4
```

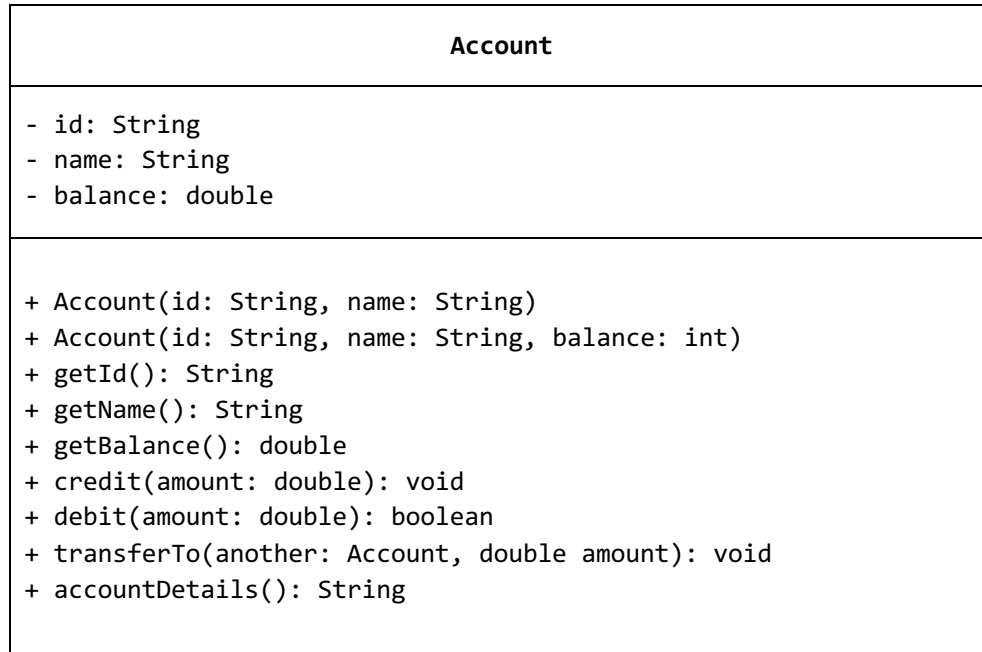
```
P2 new coordinates are:
```

```
x = 2
```

```
y = 3
```

```
The distance from (1, 4) to (2, 3) is: 1.4142135623730951
```

Problem 3:



Given the UML diagram above, create a java class named Account, the methods in the Account class perform the following tasks:

- `credit()`: Adds the value of the argument amount to the balance.
- `debit()`: Subtracts the value of the argument amount from the balance and returns true, if no enough funds are available, the method prints "Amount exceeded", then returns false.
- `transferTo()`: Transfers funds from this Account object to the parameter Account object, the amount of the transfer should be equal to the value of the argument amount.
Make sure the transfer does not cause an overdraft. Hint: Take advantage of the `debit()` method you just completed.
If the transfer was successfully completed, the method prints "Transfer completed", otherwise the method prints "Amount exceeded".

- `accountDetails()`: Returns a String that contains the Account id, name and balance in the following format:

John, the current balance in your account S93203 is \$2934.31

Where John is the name, S93203 is the account id and 2934.31 is the balance.

- **Your output should look exactly the same as follows:**

```
S93203
John
2534.31
Amount exceeded
2534.31
2334.31
John, the current balance in your account S93203 is $2934.31
-----
Amount exceeded
Transfer completed
John, the current balance in your account S93203 is $2434.31
Sarah, the current balance in your account C45442 is $840.97
-----
```

Problem 4:

Complete the `isExactReverse()` method in `Reverse.java` as follows:

- The method takes two Strings `x` and `y` as parameters and returns a `boolean`.
- The method determines if the String `x` is the exact reverse of the String `y`.
- If `x` is the exact reverse of `y`, the method returns `true`, otherwise, the method returns `false`.
- Other than uncommenting the code, do not modify the main method in `Reverse.java`.
- Sample runs provided below.

Argument String <code>x</code>	Argument String <code>y</code>	Return Value
"ba"	"a"	false
"desserts"	"stressed"	true
"apple"	"apple"	false
"regal"	"lager"	true
"war"	"raw"	true
"pal"	"slap"	false

Problem 5:

Complete the `seperateDuplicateChars()` method in `SeperateDuplicates.java` as follows:

- The method takes a `String str` as a parameter and returns a new `String`.
- The returned `String` should be exactly like `str`, but any identical characters that appear in a consecutive way must be separated by hyphens “-”.
- You may **ONLY** use the following methods from the `String` class: `charAt()`, `substring()` and `length()`.
- Other than uncommenting the code, do not modify the main method in `SeperateDuplicates.java`.
- Sample runs provided below.

Argument String str	Return String
“Hello”	“Hel-lo”
“Bookkeeper”	“Bo-ok-ke-eper”
“Yellowwood door”	“Yel-low-wo-od do-or”
“Chicago Cubs”	“Chicago Cubs”

What to turn in:

There should be 5 **.java** files(Clothing.java, Point.java, Account.java, Reverse.java and SeperateDuplicates.java), put all those files into a zip file and name it <YourFirstName_YourLastName>.zip, submit the zip file into the Dropbox on D2L.

How to zip multiple files?

On Windows: Select all the files > right click > Send to > Comprised File

On Mac: Select all the files > Click/Tap with two fingers > Compress Items