Lab - Inheritance and Composition APCS A

For the following problems, which of these problems are examples of *inheritance*? Which of these problems are examples of *composition*? Answer these questions in the top comments of your code.

1] Write a class called `MonetaryCoin` that is derived from the `Coin` class that includes a field called `value`. The `Coin` class can be downloaded from the classes *repo*. Make accessor methods and modifier (mutator) methods for this new field. Make a program `MonetaryCoinRunner.java` that uses the `MonetaryCoin` class to store the value of the coin and flip it five times. This program needs to read in the value of the coin, flip it five times showing if it is heads or tails and then outputs the coin's value. Use the `toString()` method of `Coin` to output if it is heads or tails. *Turn in these two files.* Here is a sample output:

```
mrdagler: java MonetaryCoinRuner
Enter the coin's value: .50
The coin is currently facing: Heads.

I just flipped the coin ...
The coin is currently facing: Heads.

I just flipped the coin ...
The coin is currently facing: Heads.

I just flipped the coin ...
The coin is currently facing: Tails.

I just flipped the coin ...
The coin is currently facing: Tails.

I just flipped the coin ...
The coin is currently facing: Heads.

The value of the coin is: $0.5
```

2a] Create a class called `Store` with the fields `costOfGoods`, `costOfEmployees`, and `revenue`. Make accessor methods and modifier (mutator) methods for all of these fields. This class needs to also include a method `getProfit` that returns the profit of the store. Test this class with `StoreRunner.java` that was made by the teacher. Here is a sample output:

```
mrdagler: java StoreRunner
What is the total cost of the goods: 500
What is the total cost of the employees: 10000
What is the total revenue of the store: 40000

The summary for the store:
  Cost of Goods: $500.0
  Cost of Employees: $10000.0
  Store's Revenue: $40000.0
  Store's Profit: $29500.0
```

2b] Make a sub-class of `Store` called `CandyStore` that includes the fields `numCandyBarsSold` and `costOfCandyBar` with accessor methods and modifier (mutator) methods for them. Inside of `CandyStore` make a new `getProfit` method that returns the profit of the candy store. This is called *Method Overriding*. Test this class with `CandyStoreRunner.java` that was made by the teacher. *Turn in the three java files from 2a and 2b.* Here is a sample output:

```
mrdagler: java CandyStoreRunner
What is the total cost of the goods: 70
What is the total cost of the employees: 1000
What is the cost of one candy bar: 2
How many candy bars were sold: 1000

  Store's Profit: $930.0
```

3a] Create a class called `Cyclist.java` that keeps track of someone who rides a bike: their name, type of bike they ride, and their average speed. Create accessor methods and modifier (mutator) methods for all of these attributes. Also, include the default constructor and a method that reads in the distance of a race and outputs how long it would take the cyclist to finish it.

3b] Create a class `BikeRace.java` that has an array of 10 `Cyclist`s and a constructor that sets the name and distance of the race. This class must also have two additional methods `addCyclist(name, bikeType, avgSpeed)` and `displayFinishOrder()`. If the bike race has 9 or fewer cyclists, the method `addCyclist` will add a new cyclist to the race. Otherwise, addCyclist needs to output an error and exit the program using `System.exit(1)`. The method `displayFinishOrder` needs to return a `String` of the order in which each cyclist finished the race using `selection sort` or `insertion sort`. Test your code using `BikeRaceRunner.java` that was made by your teacher. *Turen in all three java files from 3a and 3b.* Here is a sample output:

```
mrdagler: java BikeRaceRunner
Enter the name of the bike race: Wildcat Up Hill
Enter the distance of the race: 100

Enter cyclists name:
Dave
Enter the type of bike the cyclist is riding:
BMX
Enter his/her speed:
17
Enter another cyclist into the race? [Y/N]
Y

Enter cyclists name:
Lisa
Enter the type of bike the cyclist is riding:
RoadBike
Enter his/her speed:
23
Enter another cyclist into the race? [Y/N]
Y

Enter cyclists name:
Adam
Enter the type of bike the cyclist is riding:
tricycle
Enter his/her speed:
8
Enter another cyclist into the race? [Y/N]
N

The Finishing Times for Wildcat Up Hill

Place           Cyclist                 Time
-------------------------------------------------------------
1               Lisa            4.3478260869565215
2               Dave            5.882352941176471
3               Adam            12.5
```