**Inheritance related Problem Statements**

Consider a superclass PurchaseItem which models customer’s purchases. This class has:

* two private instance variables name (String) and unitPrice (double).
* One constructor to initialize the instance variables.
* A default constructor to initialize name to “no item”, and unitPrice to 0. use this()
* A method getPrice that returns the unitPrice.
* Accessor and mutator methods.
* A toString method to return the name of the item followed by @ symbol, then the unitPrice.

Consider two subclasses WeighedItem and CountedItem. WeighedItem has an additional instance variable weight (double) in Kg while CountedItem has an additional variable quantity (int) both private.

- Write an appropriate constructor for each of the classes making use of the constructor of the superclass in defining those of the subclasses.

- Override getPrice method that returns the price of the purchasedItem based on its unit price and weight (WeighedItem), or quantity (CountedItem). Make use of getPrice of the superclass

- Override also toString method for each class making use of the toString method of the superclass in defining those of the subclasses.

toString should return something that can be printed on the receipt.

For example

Banana @ 3.00 1.37Kg 4.11 SR (in case of WeighedItem class)

Pens @ 4.5 10 units 45 SR (in case of CountedItem class)

Write an application class where you construct objects from the two subclasses and print them on the screen.

1. Write an inheritance hierarchy for classes Quadrilateral, Trapezoid, Parallelogram,

Rectangle and Square. Use Quadrilateral as the superclass of the hierarchy. Create and use a Point class to represent the points in each shape. Make the hierarchy as deep (i.e., as many levels) as possible. Specify the instance variables and methods for each class. The private instance variables of Quadrilateral should be the *x-y* coordinate pairs for the four endpoints of the Quadrilateral.

Write a program that instantiates objects of your classes and outputs each object’s area (except Quadrilateral)

1. Using the Account class as a base class, write two derived classes  SavingsAccount and CurrentAccount. A SavingsAccount object, in addition to the attributes of an Account object, should have an interest variable and a method which adds interest to the account. A CurrentAccount object, in addition to the attributes of an Account object, should have an overdraft limit variable. Ensure that you have overridden methods of the Account class as necessary in both derived classes.

Now create a Bank class, an object of which contains an array of Account objects. Accounts in the array could be instances of the Account class, the SavingsAccount class, or the CurrentAccountclass. Create some test accounts (some of each type).

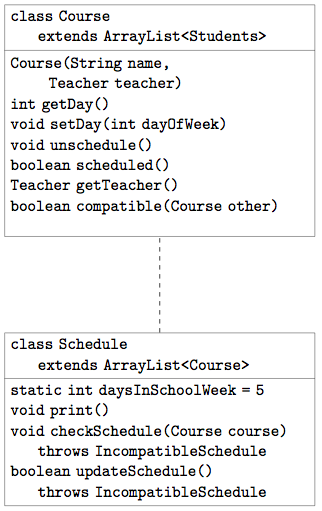
Write an update method in the bank class. It iterates through each account, updating it in the following ways: Savings accounts get interest added (via the method you already wrote); CurrentAccounts get a letter sent if they are in overdraft.

The Bank class requires methods for opening and closing accounts, and for paying a dividend into each account.

Hints:

* Note that the balance of an account may only be modified through the deposit(double) and withdraw(double) methods.
* The Account class should not need to be modified at all.
* Be sure to test what you have done after each step.

1. Course and Schedule hierarchy



Create a class Course. Each course has a single Teacher and many students.

* Think about how to add students to the course. What do you need to do that make that possible?
* Each course is either scheduled or not scheduled for a specific day of the week, which we can represent by an int.
* The setDay(int) method schedules a course for a given day of the week.
* The method compatible checks if two courses can co-exist at the same time (that is, they do not share any People in common).

Next, we are going to write a class Schedule that tries to find a compatible schedule for a list of courses.

**Problem Statement**

Write the code to implement the concept of inheritance forVehicles. You are required to implement inheritance betweenclasses. You have to write four classes in java i.e. one superclass, two sub classes and one driver class.

Vehicle is the super class whereas Bus and Truck are sub classesof Vehicle class. Transport is a driver class which contains mainmethod.

**Detailed description:**

Detailed description of Vehicle (Super class):

The class Vehicle must have following attributes:

1. Vehicle model
2. Registration number
3. Vehicle speed (km/hour)
4. Fuel capacity (liters)
5. Fuel consumption (kilo meters/liter)

The Vehicle class must have following methods:

1. Parameterized constructor that will initialize all the data members with the given values.
2. Getters and Setters for each data member that will get and set the values of data members of class.
3. A method ***fuelNeeded()*** that will take ***distance (in kilo meter)*** as an argument. It will calculate the amount of fuel needed for the given distance and will return the value of fuel needed for given distance. You can use the attributes ‘*Fuel consumption’* defined within above Vehicle class to determine the fuel needed for the given distance. You are required to implement this functionality by yourself.
4. A method ***distanceCovered()*** that will take ***time*** (in hours) as an argument. It will calculate the distance for the given time and speed and returns the value of distance. The formula to calculate speed is given as **speed = distance/time**. You can use this formula to calculate the distance.
5. A ***display()*** method that will display all the information of a vehicle.

Detailed description of Truck (Sub class):

The class Truck must have following attribute:

Cargo weight limit (Kilo grams)

The above class must have following methods:

1. Parameterized constructor that will initialize all data memberswith the given values.
2. Getters and setters for each data member that will get and setthe values of data members of class.
3. It must also override the ***display()***method of Vehicle class and must call display() method of superclass within overridden method.

Detailed description of Bus (Sub class):

The class Bus must have following attribute:

No of passengers

The above class must have following methods:

1. Parameterized constructor that will initialize all the datamembers with given values.
2. Getters and setters that will get and set the value of eachdata member of class.
3. It must also override the ***display()***method of Vehicle class and must call display method of super classwithin overridden method.

Create a class ***Transport*** which contains the main method. Perform the following within mainmethod:

* Create an instance of class Truck and initialize all the datamembers with proper values.
* Create an instance of class Bus and initialize all the datamembers with proper values.
* Now, call **fuelNeeded*()***,***distanceCovered****()* and***display()*** methods using objects of theseclasses.

Write a Java Application to define a super class Vehicle. Make two classes Bus and Truck which inherits Vehicle class. Vehicle class is having data members such as Model number, Registration number, Vehicle speed (km/hour), Fuel capacity (liters) and Fuel consumption (kilo meters/liter). Vehicle class also defines one member method named  fuelNeeded() that will takedistance (in kilo meter) as an argument. It will calculate the amount of fuel needed for the given distance and will return the value of fuel needed for given distance. Bus class is having one data member number of passengers. Truck class is having one data member cargo weight limit (kilograms).

Write class definitions, the constructors, set methods, get methods for all classes. Supply a test program that creates some instances of the Bus class and Truck class and tests all the methods.

OR

Define a class named “Vehicle” having data members such as Model number, Registration number, Vehicle speed (km/hour), Fuel capacity (liters) and Fuel consumption (kilo meters/liter). Vehicle class also defines one member method named  fuelNeeded() that will takedistance (in kilo meter) as an argument. It will calculate the amount of fuel needed for the given distance and will return the value of fuel needed for given distance. Derive another class named “Bus” from Vehicle class having one data member number of passengers.

Write a Java Application to implement a super class Person. Make two classes Student and Faculty which inherits Person class