**Question-1**

The class Atts holds a few attributes and has a method called asString. The attributes are name and color -both are strings and amount which is an int.Their visibility is public. asString returns a string showing all the Atts in a format.

for example:

Atts a = new Atts();  
a.name = "table";  
a.color = "brown";  
a.amount = 1;  
  
System.out.println(a.asString());

name: table color: brown amount: 1

Create the required attributes and method inside the atts class

**Question-2**

The LameCalculator class has instance methods that can do simple math operations on two numbers. The methods are: plus, minus, multiply and divide.

All the methods get two ints then the method does the required math operation and returns the result as an int.

for example:

LameCalculator lc = new LameCalculator();

lc.plus(1,1)  
returns:2

lc.minus(1,1)  
returns:0

lc.multiply(2,1)  
returns:2

lc.divide(10,2)  
returns:5

**Question-3**

In Db class we have two attributes/instance variables with the private visibility. meaning they cannot be accessed outside of the class.

Instance variables:  
1) name: data -  
type: String  
Visibility/Access modifier: private  
2) name: yint -  
type: int  
Visibility/Access modifier: private

Methods:  
- insertData - lets you set the values to above encapsulated instance variables.  
- getData - getter method for private data  
- getYint - getter method for private yint  
- setData - setter method for private data  
- setYint - setter method for private yint

for example

Db db = new Db();  
db.insertData("aaa",123);  
  
System.out.println(db.getData());//aaa  
System.out.println(db.getYint());//123  
  
db.setData("zzz");  
db.setYint(200);  
  
System.out.println(db.getData());//zzz  
System.out.println(db.getYint());//200

**Question-4**

Person class has following attributes:

- private String instance variables: firstName, lastName  
- private int instance variable: age

Methods:  
- getter and setter methods for each instance variable. try to use this. keyword  
- toString method.No parameters and Returns(does not print! no println in the method!) person info in this format: "firstName | lastName | age"

Constructors:

1) No-Args constructor  
-sets default values for the Person object  
name and lastName => "undefined"  
age => -1

2) 3-Args Constructor:  
- accepts firstName, lastName, age parameters and assigns values to encapsulated instance variable  
try to use this. keyword

Examples:

Person person = new Person();

System.out.println(person.getFirstName()); //"undefined"  
System.out.println(person.getLastName()); //"undefined"  
System.out.println(person.getAge()); // -1  
System.out.println(person.toString()); //"undefined | undefined | -1"

person.setFirstName("John");  
person.setLastName("Doe");  
person.setAge(44);

System.out.println(person.toString() ); // "John | Doe | 44"

Person person = new Person("Fatima", "Lee", 22);

System.out.println(person.getFirstName()); //"Fatima"  
System.out.println(person.getLastName()); //"Lee"  
System.out.println(person.getAge()); // 22  
System.out.println(person.toString()); //"Fatima | Lee | 22"

**Question-5**

Write a custom class TV that has 4 instance variables: int channel=1, int volumeLevel=1, boolean on = false, String brand = "undefined".  
Create one "no arguments" constructor that prints message: "Creating TV object using no Args- constructor".  
Also, create one more constructor that defines the instance variable brand, and displays the message: "Creating TV object using 1 arg - constructor".  
Generate getters/setters for volumeLevel (getVolumeLevel/setVolumeLevel), channel (getChannel/setChannel) and brand (getBrand/setBrand) instance variables.  
Declare methods channelUp() to increase variable channel by one, and channelDown() to decrease variable channel by one.  
Declare methods volumeUp() to increase variable volume by one, and volumeDown() to decrease variable volume by one.  
Create isOn(), turnOn() and turnOff() methods for "on" instance variable. If the tv is already on, no need to turn it on again, display the message: "TV is already ON". If the tv is already off, then no need to turn it off again, display the message: "TV is already OFF".  
The value of a variable channel cannot be negative or zero, and cannot be higher than 120. If the user will try to set an invalid channel, display the message: "ERROR: TV is either OFF or invalid Channel".  
Volume can be only in the range between 1 and 7. You may change volume only if TV is on. Otherwise, display the message: "ERROR: TV is either OFF or invalid Volume level". Create isOn() method that will check tv status.

**Question-6**

Write a class named ParkingMeter containing:  
Two instance variables named timeLeft and maxTime of type int. The value of timeLeft should be initialized to 0.  
A constructor accepting a single integer parameter whose value is used to initialize the maxTime instance variable.  
A method named add that accepts an integer parameter. If the value of the parameter is equal to 25, the value of timeLeft is increased by 30; otherwise, no increase is performed. Furthermore, the increase occurs only if the value of timeLeft will not exceed the value of maxTime. add returns a boolean value: true if timeLeft was increased, false otherwise.  
A method named tick that accepts no parameters and returns no value. tick decreases the value of timeLeft by 1, but only if the value of timeLeft is greater than 0.  
A method named isExpired that accepts no parameters. isExpired returns a boolean value: true if the value of timeLeft is equal to 0; false otherwise.

**Question-7**

Write a class named GasTank containing:  
An instance variable named amount of type double, initialized to 0.  
An instance variable named capacity of type double.  
A constructor that accepts a parameter of type double. The value of the parameter is used to initialize the value of capacity.  
A method named addGas that accepts a parameter of type double. The value of the amount instance variable is increased by the value of the parameter. However, if the value of amount is increased beyond the value of capacity, amount is set to capacity.  
A method named useGas that accepts a parameter of type double. The value of the amount instance variable is decreased by the value of the parameter. However, if the value of amount is decreased below 0, amount is set to 0.  
A method named isEmpty that accepts no parameters. isEmpty returns a boolean value: true if the value of amount is less than 0.1, and false otherwise.  
A method named isFull that accepts no parameters. isFull returns a boolean value: true if the value of amount is greater than capacity-0.1, and false otherwise.  
A method named getGasLevel that accepts no parameters. getGasLevel returns the value of the amount instance variable.  
A method named fillUp that accepts no parameters. fillUp increases amount to capacity and returns the difference between the value of capacity and the original value of amount (that is, the amount of gas that is needed to fill the tank to capacity).

**Question-8**

Write a class definition of a class named 'Value' with the following:  
a constructor accepting a single integer parameter  
a constructor with no parameters  
a method 'setVal' that accepts a single parameter,  
a boolean method, 'wasModified' that returns true if setVal was ever called for the object.  
a method 'getVal' that returns an integer value as follows: if setVal has ever been called, it getVal returns the last value passed to setVal. Otherwise if the "single int parameter" constructor was used to create the object, getVal returns the value passed to that constructor. Otherwise getVal returns 0.

**Question-9**

Write the definition of a class Telephone.  
The class has no constructors, one instance variable of type String called number, and two public static variables.  
One is of type int called quantity, initialized to 250;  
the other is of type double called total, initialized to 15658.92.  
Create getNumber() and setNumber() methods that will manipulate with number instance variable.

**Question-10**

A static method can be used without instantiating a class object. Like in the calculator class create a plus and minus. but this time make them static methods. plus: gets two ints and returns the addition of the to numbers as an int   minus: the same as plus its also static, but it returns the substructed value of the two ints it gets

calc.minus(1,1)  
returns:0

calc.plus(10,1)  
returns:11

**Question-11**

Assume the existence of a BankAccount class.  
Define a subclass, SavingsAccount that contains the following: A double instance variable, interestRate.  
A method getInterestRate that returns a value of interestRate.  
A constructor that accepts a parameter of type double which is used to initialize the instance variable.

**Question-12**

Assume the existence of a Phone class that is a super class for CameraPhone class.  
Define a subclass, CameraPhone that contains two instance variables: an integer named imageSize, representing the size in megapixels (for simplicity assume a pixel takes up one byte-- thus megapixels equals megabytes) of each picture (i.e., 2 means each image is composed of 2 megapixels), and an integer named memorySize, representing the number of gigabytes in the camera's memory (i.e., 4 means 4 Gigabytes of memory).  
There is a constructor that accepts two integer parameters corresponding to the above two instance variables and which are used to initialize the respective instance variables.  
There is also a method named numPictures that returns (as an integer) the number of pictures the camera's memory can hold. Don't forget that a gigabyte is 1,000 megabytes.ool