Lab Session 3 : Objects and Classes

Exercise 1: Write a Java class "Cow" and a Java tester class "CowTestDrive" to implement the class design: "Cow"

- Apply "Encapsulation" concept in your classes
- The moo() method should print out the text "Moo…!"
- The tester class should:
    - Make a Cow object
    - Set the age of the Cow to 4
    - Call the moo() method

Cow
name breed age
moo()

Exercise 2: Write a Java class "NameCard" and a Java tester class "NameCardTestDrive" to implement the NameCard design:

- Apply "Encapsulation" concept in your classes
- Print demo results to the screen


```
Name  Polly Morfism
Phone  555-0343
eMail  pm@wickedlysmart
```

Exercise 3: Write a Java class "Vector" and a Java tester class "VectorTestDrive" to implement the class design: "Vector".

- Apply "Encapsulation" concept in your classes.
- In your tester class "VectorTestDrive":
    - Create and print out information of 2 vectors
    - Calculate and print out sum, subtraction and multiplication of the two created vectors

| Vector |
| --- |
| int x |
| int y |
| add(Vector other) |
| subtract(Vector other) |
| multiply(Vector other) |

Exercise 4: Write a Java class "ShoppingCart" and a Java tester class "ShoppingCartTestDrive" to implement the class design: "ShoppingCart".

- Apply "Encapsulation" concept in your classes
- Implement three methods addToCart(), removeFromCart(), checkOut() and print demo results to the screen
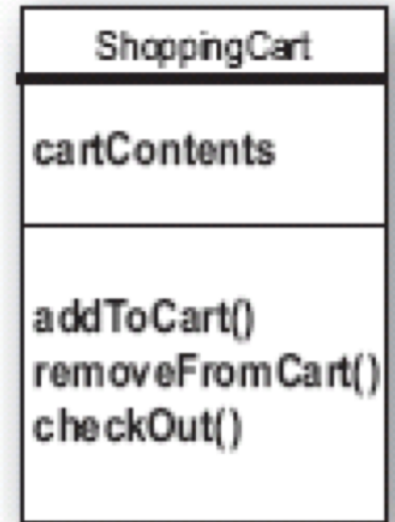
Exercise 5: Write a Java class "Button" and a Java tester class "ButtonTestDrive" to implement the class design: "Button".

- Apply "Encapsulation" concept in your classes
- Implement four methods setColor(), setLabel(), dePress(), unDepress() and print demo results to the screen
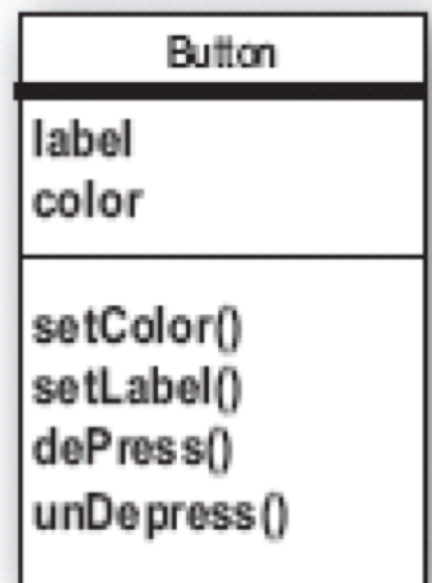
Exercise 6: The following codes implement the OOP class design: "Automobile". However, these codes have **some problems with OOP principles**. Try to run these codes, then find the problems and fix them as much as possible.

class Automobile

Automobile
- fuel: double
- speed: double
- license: String
+ accelerate (double pedalPressure): void
+ decelerate (double pedalPressure): void

```
class Automobile {
    static double fuel;
    static double speed;
    static String license;
    static void init(double f, double s, String l) {
        fuel = f;
        speed = s;
        license = l;
    }
    static void accelerate(double v) {
        if (fuel > 0) speed += v;
    }
    static void decelerate(double v) {
        if (fuel <= 0) speed -= v;
    }
    public static void main(String args[]) {
        init(4.5,34,"29JAD");
        accelerate(4);
        decelerate(5);
    }
}
```