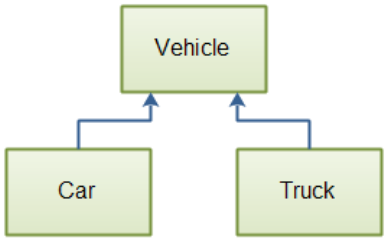
OOP Java Lab Tasks 7: Access Modifiers, Inheritance, TypeCasting

Task-1: Class Level Attributes (static attributes):

Fields/Methods that are declared as ***static*** are shared by all instances of that class. Consider a system to maintain the information of all employees at the company. Create a class *Employee* to store the information of all employees. Assign a unique ID to each incoming employee. Total number of employees should be known to all employees working at the company. Think about how we can share this information.

Task-2: Vehicle Example Inheritance




```
public class Vehicle {
    protected String licensePlate = null;

    public void setLicensePlate(String license) {
        this.licensePlate = license;
    }
}
```

- Design child class Card and Truck with an additional field of your own choice.

Task-3: Inheritance (Banking System Example)

Fields/Methods that are declared as ***protected*** in the parent class are also visible to all instances of that child class. In the example of the BankAccount below, think about if all fields / methods that need to be declared *protected* or for methods thus to be overridden.
In last lab, we designed a class called ***BankAccount*** that allowed deposits and withdraws to an account. Following on that, lets extend to a complete Banking System.

Start a new project called BankManager that will contain the following classes:

CheckingAccount
SavingsAccount
CertificateOfDeposit

Make sure they all extend from the same class called BankAccount that includes all the common fields. For each type of account, think about the details of how they function. What fields need to be declared in the parent class and what fields/methods should be specific to the child classes.

Follow these steps and check them once you're done to complete this exercise:

1. Create a new class called BankAccount with account and balance fields.
2. Create *deposit()* and *withdraw()* methods to make changes to account balance.
3. Override a toString() method to display the fields of class BankAccount.
4. Create a new class called CheckingAccount that extends BankAccount with an extra field called as limit.
5. Repeat the same for SavingsAccount and CertificateOfDeposit as in Step.2 above.
6. For child classes, override methods of parent classes as necessary.

In the main method, create an instance of each of the 3 child classes. Make sure you can access the account and balance fields (set them and read them) for all account types.

Good Luck!