

Section 1 - Inheritance:

- Create a class called `Vehicle` with properties `make`, `model`, `year`, `color`, and a method `drive()`.
- The `drive()` method should log "Driving {make} {model}." to the console.
- Create a subclass called `Car` that extends `Vehicle` and adds a property `numSeats`.
- Create a subclass called `RideShareCar` that extends `Car` and adds a property `isAvailable`.

Section 2 - Polymorphism:

Create a class called `Shape` with a method `calculateArea()`. Then, create three subclasses called `Rectangle`, `Triangle`, and `Circle`, each of which extends `Shape`.

Each subclass should override the `calculateArea()` method as follows:

- `Rectangle` should accept `width` and `height` parameters and calculate the area as `width * height`.
- `Triangle` should accept `base` and `height` parameters and calculate the area as `(base * height) / 2`.
- `Circle` should accept `radius` parameter and calculate the area as `Math.PI * radius * radius`.

Finally, create instances of each subclass and call the `calculateArea()` method on each instance, logging the result to the console.

Section 3 – Abstraction and encapsulation:

Create a class called `BankAccount` with properties `accountNumber`, `balance`, and `accountHolderName`. Then, create two subclasses called `CheckingAccount` and `SavingsAccount`, each of which extends `BankAccount`.

Make the `accountNumber`, `balance`, and `accountHolderName` properties private by using the `#` symbol before their names.

Implement the `deposit()` and `withdraw()` methods in the `CheckingAccount` and `SavingsAccount` subclasses. These methods should accept a `amount` parameter and update the `balance` property accordingly.

However, for the `SavingsAccount` subclass, if the balance drops below zero after a withdrawal, the withdrawal should fail and the balance should not be updated.

Implement the `getBalance()` method in both the `CheckingAccount` and `SavingsAccount` subclasses. This method should return the `balance` property.

Finally, create instances of each subclass and call the `deposit()`, `withdraw()`, and `getBalance()` methods on each instance, logging the result to the console.