### 1. Scenario: Library System

You are creating a system to manage books in a library. Each book has a title, author, and ISBN. The system needs to allow updating the book’s availability status (whether it’s checked out or not) and updating its title.

* **Question**:  
  How would you define a Book class with properties title, author, and isbn?
  + What methods would you include to:
    - Change the availability status (e.g., checkOut() and returnBook() methods)?
    - Update the book’s title (e.g., updateTitle(newTitle) method)?

**Code**

//class

class Book {

    title;

    author;

    isbn;

    //constructor

    constructor(title, author, isbn) {

        this.title = title;

        this.author = author;

        this.isbn = isbn;

    }

    //methods

    checkout() {

        console.log(this.title, 'checked');

    }

    returnBook() {

        console.log(this.title, 'returned');

    }

    updateTitle(newTitle) {

        this.title = newTitle;

    }

}

//object

var B1 = new Book("manifesting", 'Allen', '2001');

console.log(B1);

B1.checkout();

B1.returnBook();

B1.updateTitle('new beginning');

console.log(B1);

### 2. Scenario: Product with Discount

You are creating an e-commerce platform, and each product has a name, price, and category. You need to be able to apply a discount to the product and retrieve its final price after applying the discount.

* **Question**:  
  How would you define a Product class with properties name, price, and category?
  + What methods would you include to:
    - Apply a discount to the product (e.g., applyDiscount(amount) method)?
    - Retrieve the price after discount (e.g., getPriceAfterDiscount() method)?

**Code:**

class Product {

    name;

    price;

    category;

    constructor(name, price, category) {

        this.name = name;

        this.price = price;

        this.category = category;

    }

    applyDiscount(amount) {

        console.log(this.name, "discount is", this.price - amount);

    }

    getPriceAfterDiscount(price) {

        console.log(this.name, "the Price after discount is", this.price - price);

    }

}

var P1 = new Product("footware", 2000, "women");

console.log(P1);

P1.applyDiscount(2000 / 10);

P1.getPriceAfterDiscount(2000 / 10);

### 3. Scenario: Bank Account System

You are designing a banking system. Each account has an account holder name, account number, and balance. You need to be able to deposit, withdraw, and check the balance of the account.

* **Question**:  
  How would you define a BankAccount class with properties accountHolderName, accountNumber, and balance?
  + What methods would you include to:
    - Deposit money into the account (e.g., deposit(amount) method)?
    - Withdraw money from the account (e.g., withdraw(amount) method)?
    - Check the current balance (e.g., getBalance() method)?

**Code :**

class BankAccount {

    accountHolderName;

    accountNumber;

    balance;

    constructor(accountHolderName, accountNumber, balance) {

        this.accountHolderName = accountHolderName;

        this.accountNumber = accountNumber;

        this.balance = balance;

    }

    deposit(amount) {

        console.log("the deposited amount in", this.accountNumber, "is", amount);

        this.balance = this.balance + amount;

        console.log("the current balance after deposit is", this.balance);

    }

    withdraw(amount) {

        console.log("withdrawn amount is", amount);

        this.balance = this.balance - amount;

        console.log("Balance after withdraw is", this.balance);

    }

    getBalance() {

        console.log("your account balance", this.balance);

    }

}

var Balance = new BankAccount("Kevin", "12345", 1000);

console.log(Balance);

Balance.deposit(1000);

Balance.withdraw(300);

Balance.getBalance();

### 4. Scenario: Vehicle Management

You are managing a fleet of vehicles. Each vehicle has a model, license plate, and mileage. You need to track the mileage whenever the vehicle is driven.

* **Question**:  
  How would you define a Vehicle class with properties model, licensePlate, and mileage?
  + What methods would you include to:
    - Simulate driving the vehicle (e.g., drive(miles) method, which increases the mileage)?
    - Get the current mileage of the vehicle (e.g., getMileage() method)?

**Code**

// code-4

class Vehicle {

    model;

    licensePlate;

    mileage;

    constructor(model, licensePlate, mileage) {

        this.model = model;

        this.licensePlate = licensePlate;

        this.mileage = mileage;

    }

    drive(miles) {

        console.log("The bike mileage has incresed by", miles, "km/ltr");

        this.mileage = this.mileage + miles;

    }

    getMileage() {

        console.log("the current mileage of this", this.model, "is", this.mileage, "km/ltr.");

    }

}

var vehicle = new Vehicle("Passion Plus", "2812", 40);

console.log(vehicle);

vehicle.drive(5);

vehicle.getMileage();

### 5. Scenario: Grading System

You are creating a system for students where each student has a name and a grade for a subject. The system should allow updating the student’s grade.

* **Question**:  
  How would you define a Student class with properties name and grade?
  + What methods would you include to:
    - Set or update the grade for the student (e.g., setGrade(newGrade) method)?
    - Retrieve the student’s current grade (e.g., getGrade() method)?

**Code**

// code -5

class Student {

    name;

    grade;

    constructor(name, grade) {

        this.name = name;

        this.grade = grade;

    }

    setGrade(newGrade) {

        this.grade = newGrade;

    }

    getGrade(grade) {

        console.log("ashish got", this.grade, "in maths");

    }

}

var student1 = new Student("Ashish", "B");

student1.getGrade();

student1.setGrade("A");

student1.getGrade();