Encapsulation

🎯 Direct Challenges

1. Create a class with private fields and public getters/setters.

```
class Person {
    private String name;
    private int age;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
}

public class Main {
    public static void main(String[] args) {
        Person p = new Person();
        p.setName("John");
        p.setAge(25);
        System.out.println("Name: " + p.getName());
        System.out.println("Age: " + p.getAge());
```

```
    }
}
```

Output:

Name: John

Age: 25

2. Demonstrate how encapsulation protects data.

Explanation: By making fields private and using setters/getters, data access is controlled and secure. External classes can't modify data directly.

3. Prevent setting negative values to age field using validation in setter.

```
public void setAge(int age) {
    if(age > 0) {
        this.age = age;
    } else {
        System.out.println("Invalid age.");
    }
}
```

Output:

Invalid age. (if age is set to a negative number)

🧩 **Scenario-Based Challenges**

1. Create a Patient class that restricts direct access to medicalHistory.

```
class Patient {
    private String medicalHistory;

    public String getMedicalHistory() {
        return "Access Denied";
```

```
    }

    public void setMedicalHistory(String history) {
        this.medicalHistory = history;
    }
}

public class Main {
    public static void main(String[] args) {
        Patient p = new Patient();
        p.setMedicalHistory("Diabetes");
        System.out.println(p.getMedicalHistory());
    }
}
```

Output:

Access Denied

2. Build a LoanAccount class and encapsulate the balance with setter restrictions.

```
class LoanAccount {
    private double balance;

    public double getBalance() {
        return balance;
    }

    public void setBalance(double amount) {
        if(amount >= 0) {
            balance = amount;
```

```
        } else {
            System.out.println("Negative balance not allowed.");
        }
    }
}

public class Main {
    public static void main(String[] args) {
        LoanAccount acc = new LoanAccount();
        acc.setBalance(-500);
        acc.setBalance(10000);
        System.out.println("Balance: " + acc.getBalance());
    }
}
```

Output:

Negative balance not allowed.

Balance: 10000.0