# Exercise OOP cơ bản

## Bài tập 1: Sử dụng PHP OOP để giải quyết bài toán sau

Sử dụng PHP OOP để khởi tạo một class Employee bao gồm các thuộc tính sau

* id
* name
* workingHoursPerDay = 8
* hourlyRate = 15
* totalLeaveTaken
* workingDays

Khởi tạo 1 instance của class Employee và viết một hàm getSalaryAmount($totalDays) cho phép tính salary của Employee đó theo công thức sau:

salary = ($totalDays – totalLeaveTaken) \* workingHourPeerDay \* hourlyRate

Sau khi tính toán được salary, hãy in ra màn hình dòng chữ sau

“name has worked for workingDays and taken totalLeaveTaken leaves, name salary is salary” với salary được tính từ hàm ở trên. Thuộc tính workingDays của object được tính bằng $totalDays của hàm được định nghĩa ở trên trừ đi $totalLeaveTaken của object

Ví dụ

John, Smith has worked for 16 days and taken 4 leaves. John, Smith salary is 1920

Jacob has worked for 18 days and taken 2 leaves. Jacob salary is 2160

## Bài tập 2: Sử dụng PHP OOP và điều kiện để giải quyết bài toán sau

Khởi tạo một class Bank có các thuộc tính như sau

* accNo
* name
* balance (Tiền trong tài khoản)

Viết 2 hàm depositAmount($amount) để thêm $amount vào balance của tài khoản và viết 1 hàm deductAmount($amount) để rút tiền từ tài khoản. Kiểm tra 3 điều kiện sau

* Nếu balance < 0 thì in ra màn hình “No balance in account”
* Nếu balance < $amount thì in ra màn hình “Requested amount is greater than balance”
* Ngược lại thì balance mới = balance cũ – amount.

Cuối cùng, viết ra một hàm checkBalance() và in ra balance của object đó trong tài khoản. Khởi tạo instance của object và sử dụng depositAmount()và deductAmount() và in ra balance của object instance đó.

## Bài tập 3: Sử dụng PHP OOP để giải quyết bài toán sau

Với yêu cầu của bài tập 2, thay đổi scope của thuộc tính balance từ public -> private. In ra kết quả tương tự như bài tập 2.

Hướng dẫn: Sử dụng constructor để truyền tham số balance từ ngoài class vào.

## Bài tập 4: Sử dụng PHP OOP Inheritance để giải quyết bài toán sau

Tạo class Animal bao gồm 2 thuộc tính

* name: String
* hunt: boolean (Kiểm tra xem con thú đó có đi săn hay không) = false;

Tạo class Dog thừa kế class Animal, định nghĩa constructor cần thiết và implement một method makeSound() với nội dung là “Woff, woff”

Tạo class Tiger thừa kế class Animal, định nghĩa constructor cần thiết và implement một method makeSound() với nội dung là “Grrr, grrr”

Khởi tạo instance của class và in ra màn hình xem 2 con vật Dog và Tiger có makeSound() và doesHunting() hay không

## Bài tập 5: Sử dụng PHP OOP Interface để giải quyết bài toán sau

## Định nghĩa một interface Phone, trong đó có một hàm makeCall($number)và một hàm sendMessage($number, $message)

Tạo một class iOS implements method ở trên và in ra màn hình nội dung của hàm makeCall() và sendMessage(). Nội dung trong hàm tuỳ chọn

## Bài tập 6: Sử dụng PHP OOP Abstract class để giải quyết bài toán sau

Khai báo một class Database. Trong class này bao gồm 2 thuộc tính connection và connectionActive và 2 hàm abstract setConnection($dbName), abstract getConnection().

Tạo một class DBConnection và extends từ Database class, implement tất cả các method được định nghĩa abstract ở trên. Property connectionActive sẽ có giá trị là true nếu hàm setConnection($dbName) được gọi. Hàm getConnection() sẽ in ra giá trị của $connection trong trường hợp connectionActive là đúng, nếu $connectionActive là sai thì không in ra gì.

Yêu cầu: In ra giá trị của $connection thông qua hàm getConnection()