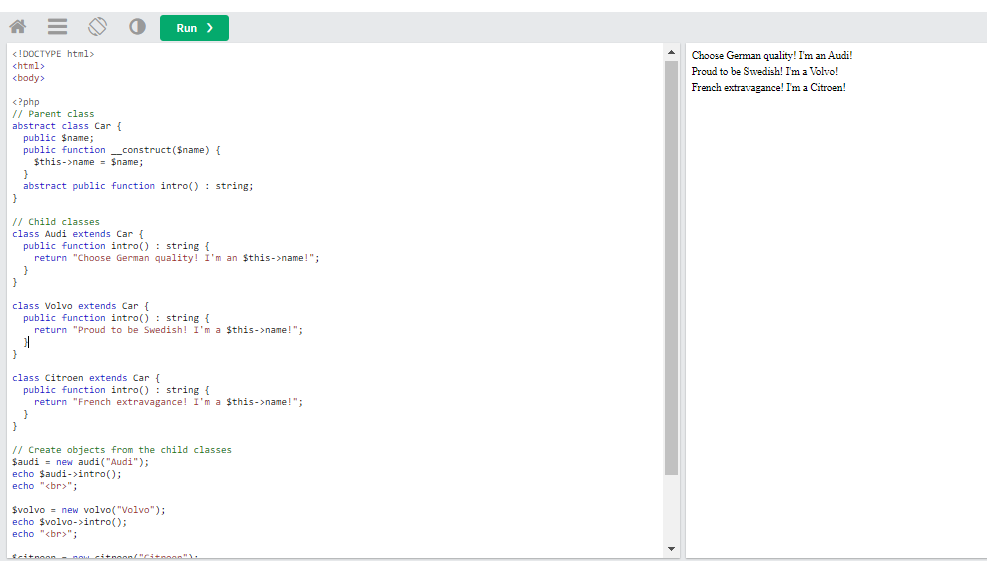
OOP

1. Abstract Classes:

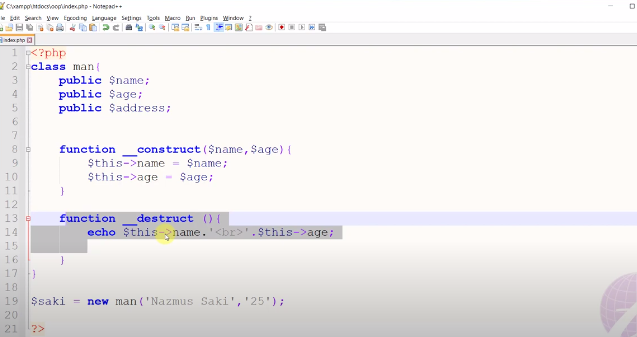
Parent ক্লাসের মধ্যে কোনো একটি মেথড ঘোষণা করেন কিন্তু সেই মেথরের কাজটি parent ক্লাস হবে না চাইল্ড ক্লাসে হবে। অর্থাৎ সে মেথরের ক্লাস executive হতে হবে যা একটি ক্লাসে



2.constructor: object create korar sathe AUTO excute hobe.call korar need hoy nai



1. Construct er kaj ses houyar pore auto destruct er vitor doke jay



3.

1. **Encapsulation ;** **মানে:** ডেটা (property) এবং ফাংশন (method) একত্রে একটি ক্লাসে প্যাক করে রাখা এবং বাইরের দুনিয়া থেকে ডেটা লুকানো।  
   **কেন:** ডেটা সুরক্ষিত রাখতে এবং শুধু method-এর মাধ্যমে access করতে দিতে।

**class BankAccount {**

**private $balance = 0; // balance সরাসরি access করা যাবে না**

**public function deposit($amount) {**

**if ($amount > 0) {**

**$this->balance += $amount;**

**}**

**}**

**public function getBalance() {**

**return $this->balance;**

**}**

**}**

**$account = new BankAccount();**

**$account->deposit(1000);**

**echo $account->getBalance(); // আউটপুট: 1000**

**ব্যাখ্যা:**  
$balance private, তাই বাইরে থেকে access করা যায় না। deposit() ও getBalance() method এর মাধ্যমে নিয়ন্ত্রণ করা হয়। এটি হলো encapsulation।

1. **Inheritance :** **মানে:** একটি ক্লাস অন্য ক্লাসের বৈশিষ্ট্য (property/method) গ্রহণ করে।  
   **কেন:** কোড পুনঃব্যবহারযোগ্য (reusable) করার জন্য।

class Animal {

public function sound() {

echo "Some generic sound";

}

}

class Dog extends Animal {

public function sound() {

echo "Bark!";

}

}

$dog = new Dog();

$dog->sound(); // আউটপুট: Bark!

🟢 **ব্যাখ্যা:**  
Dog ক্লাস Animal থেকে বৈশিষ্ট্য পেয়েছে। sound() method override করেছে।

### 3. **Polymorphism (পলিমরফিজম)**

**মানে:** একই method ভিন্নভাবে বিভিন্ন ক্লাসে কাজ করে।  
**কেন:** একাধিক ক্লাসে একই ইন্টারফেস মেইনটেইন করে behavior পরিবর্তন করা যায়।

class Animal {

public function makeSound() {

echo "Animal sound";

}

}

class Cat extends Animal {

public function makeSound() {

echo "Meow";

}

}

class Cow extends Animal {

public function makeSound() {

echo "Moo";

}

}

// ব্যবহার

function printSound(Animal $animal) {

$animal->makeSound();

}

printSound(new Cat()); // Meow

printSound(new Cow()); // Moo

🟢 **ব্যাখ্যা:**  
printSound() function একই টাইপের ইনপুট নিয়ে ভিন্নভাবে কাজ করছে। এটাই Polymorphism।

### 4. **Abstraction (অ্যাবস্ট্রাকশন)**

**মানে:** দরকারি জিনিস দেখানো এবং অপ্রয়োজনীয় জিনিস লুকানো।  
**কেন:** জটিলতা লুকিয়ে ইউজারের জন্য কাজ সহজ করা।

abstract class Shape {

abstract public function area(); // শুধু ঘোষণা, কোন বাস্তবায়ন নেই

}

class Rectangle extends Shape {

private $length;

private $width;

public function \_\_construct($l, $w) {

$this->length = $l;

$this->width = $w;

}

public function area() {

return $this->length \* $this->width;

}

}

$rect = new Rectangle(5, 4);

echo $rect->area(); // 20

**ব্যাখ্যা:**  
Shape ক্লাসে শুধু method এর নাম ঘোষণা আছে। কিভাবে কাজ করবে সেটা child class Rectangle define করেছে। এটি হলো abstraction।