**Member function and veriable**

ক্লাসের ভিতর ভেরিয়েবল কে বলা হয় প্রোপর্টিজ বা মেম্বার ভেরিয়েবল এবং ফাংশনকে বলা হয় মেথড বা মেম্বার ফাংশন বা (behaviour)বিহেবিয়ার।

\_\_construct(): পিএইচপির বিশেষ ধরণের বিল্ট-ইন মেথড। অবজেক্ট তৈরির সময় প্রোপার্টির ভ্যালু এসাইনে সম্মতিদেয়। নতুন কোনো অবজেক্ট ক্রিয়েট করলে ক্লাসের মধ্যে যে মেথডটি সবার আগে এক্সেকিউট হয়, তাকে \_\_\_কন্সট্রাক্টর বলে।

ফাংশন: · Function হল Group of Code যা একটি নির্দিষ্ট কাজ করে । এক বা একাধিক ইনপুট প্যারামিটার গ্রহণ করতে পারে । একটি আউটপুট ভ্যালু রিটার্ন করতে পারে ।

Function এবং Constructor এরপার্থক্যগুলো

• Function হল Group of Code যা একটি নির্দিষ্ট কাজ করে অন্যদিকে Constructor এর কাজ হল class এর Instance তৈরি করা।

Function এর return type থাকে কিন্তু Constructor এর কোন return type নেই এমনকি void ও ব্যবহার করা যায়না।

Function এ final, static, abstract এই ধরনের Modifier গুলো ব্যবহার হয় কিন্তু Constructor এ এই ধরনের Modifier গুলো ব্যবহার করা যায় না।

Function এ this, super কিওয়ার্ড ব্যবহার হয় Working Class এবং Parent Class এর Instance কে Refer করার জন্য অন্যদিকে constructor এ this, super কিওয়ার্ড ব্যবহার হয় Working Class এবং Parent Class এর Constructor কে Call করারজন্য।

---------------------------------

প্রোগ্রামিং ভাষায় কনটেক্সট কি ?

---------------------------------

প্রোগ্রামিং ভাষায়, কনটেক্সট বলতে একটি স্পেসিফিক এনভায়রনমেন্ট বোঝায় যেখানে কোড কীভাবে কাজ করবে, কোথায় ব্যবহৃত হবে, এবং কী ধরনের ডেটা প্রিপায়ার করা হবে । কনটেক্সট গুরুত্বপূর্ণ কারণ এটি প্রোগ্রামারদের কোড বুঝতে এবং ডিবাগ করতে সাহায্য করে ।

কনটেক্সটের উদাহরণ:

একটি ভ্যারিয়েবলের মান

একটি ফাংশনের আর্গুমেন্ট

একটি ক্লাসের অবস্থা

একটি প্রোগ্রামের ইনপুট এবং আউটপুট

---------------------------------

প্রোগ্রামিং ফিলোসফি কি ?

---------------------------------

প্রোগ্রামিং ফিলোসফি হল প্রোগ্রামিং এর প্রিন্সিপল, প্রসিডিউর(পদ্ধতি) এবং টেকনিক(কৌশল) সম্পর্কে স্পষ্ট ধারণা দেয়, যে প্রোগ্রামাররা কীভাবে কোড লিখবে এবং কীভাবে সফ্টওয়্যার ডেভেলপমেন্ট করবে । প্রোগ্রামিং ফিলোসফি প্রোগ্রামারদের ভালো কোড লিখতে এবং উন্নত সফ্টওয়্যার ডেভেলপ করতে সাহায্য করে।

প্রোগ্রামিং ফিলোসফি-র কিছু উদাহরণ: -----------------

DRY (Don't Repeat Yourself): কোড রিউস করা ।

KISS (Keep It Simple Stupid): কোড যতটা সম্ভব সিম্পল রাখার ।

SOLID: মেইনটেইনেবল এবং ফ্লেক্সিবল সফ্টওয়্যার তৈরির জন্য SOLID প্রিন্সিপল ফলো করা ।

YAGNI (You Ain't Gonna Need It): প্রয়োজন না হওয়া পর্যন্ত কোড না লেখার ।

প্রোগ্রামিং আইসোলেট কি ?

---------------------------------

প্রোগ্রামিং আইসোলেট হলো একটি প্রোগ্রামিং টেকনিক যেখানে একটি কোডকে নির্দিষ্ট স্কোপের মধ্যে আইসোলেট করে এবং অন্য কোডের সাথে ইন্টারঅ্যাকশন প্রিভেন্ট করে ।

প্রোগ্রামিং আইসোলেট ব্যবহারের কারণ: -----------------

Modularity: কোডকে মডিউল করে, মেইনটেইনএবল এবং টেস্টএবল করে তোলে।

Security: বাগ এবং সিকিউরিটি ভারনারেবিলিটির রিস্ক কমিয়ে দেয় ।

প্রোগ্রামিং আইসোলেট তৈরি করার বিভিন্ন উপায়: -----------------

Function: ফাংশনের মাধ্যমে কোডকে আইসোলেট করা যায় । একটি বেসিক ওয়ে ।

Class: ক্লাসের মাধ্যমে কোডকে আইসোলেট করা যায় । একটি পাওয়ারফুল ওয়ে ।

Module: মডিউল মাধ্যমে বড় ধরণের অ্যাপ্লিকেশনের কোডকে আইসোলেট করা যায় ।

---------------------------------

প্রোগ্রামিং প্যারাডাইম(দৃষ্টান্ত) কি ?

---------------------------------

প্রোগ্রামিং প্যারাডাইম হল প্রোগ্রাম লেখার উপায় । অর্থাৎ এটি নির্ধারণ করে যে প্রোগ্রাম কিভাবে লিখতে হবে এবং কীভাবে কাজ করে ।

বিভিন্ন ধরণের প্রোগ্রামিং প্যারাডাইম রয়েছে: -----------------

ইম্পারেটিভ প্রোগ্রামিং: এটি সবচেয়ে সাধারণ প্রোগ্রামিং প্যারাডাইম। এটি প্রোগ্রামার-কে নির্দেশ দেয় কিভাবে একটি নির্দিষ্ট কার্য সম্পাদন করতে হবে। C, C++, Java, Python

ডিক্লরেটিভ প্রোগ্রামিং: এটি প্রোগ্রামার-কে নির্দেশ দেয় কিভাবে একটি নির্দিষ্ট কার্য সম্পাদন করার জন্য ডেটা কে সংগঠিত করতে হবে। SQL, Prolog

প্রসিডিউরাল প্রোগ্রামিং: এটি প্রোগ্রামার-কে নির্দেশ দেয় কিভাবে একটি নির্দিষ্ট কার্য সম্পাদন করার জন্য ফাংশন গুলো কে সংযুক্ত করতে হবে। Haskell, Lisp

অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং: এটি প্রোগ্রামার-কে নির্দেশ দেয় কিভাবে একটি নির্দিষ্ট কার্য সম্পাদন করার জন্য অবজেক্ট গুলো কে ব্যবহার করতে হবে। Java, C++, Python

PHP প্রোগ্রামিং প্যারাডাইম: -----------------

PHP একটি মাল্টি-প্যারাডাইম প্রোগ্রামিং ল্যাংগুয়েজ, যার মানে বিভিন্ন প্রোগ্রামিং প্যারাডাইম ব্যবহার করে প্রোগ্রাম লেখ যায় । PHP তে নির্দিষ্ট কোন প্রোগ্রামিং প্যারাডাইম নেই । প্রোগ্রামার তাদের প্রয়োজন অনুযায়ী প্যারাডাইম গুলো নির্বাচন করতে পারে।

PHP এর সবচেয়ে সাধারণ প্রোগ্রামিং প্যারাডাইম হল: -----------------

প্রসিডিউরাল প্রোগ্রামিং: ফাংশন এবং প্রসিডিউর ব্যবহার করে প্রোগ্রাম লেখা হয় ।

অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং: অবজেক্ট এবং ক্লাস ব্যবহার করে প্রোগ্রাম লেখা হয় ।

---------------------------------

প্রসিডিউরাল প্রোগ্রামিং কি ?

---------------------------------

প্রসিডিউরাল প্রোগ্রামিং হলো এক ধরণের প্রোগ্রামিং প্যারাডাইম যেখানে ফাংশন এবং প্রসিডিউর ব্যবহার করে প্রোগ্রাম লেখা হয় ।

প্রসিডিউর: একটি নির্দিষ্ট কার্য সম্পাদন করে। একটি বা ততোধিক ইনপুট প্যারামিটার গ্রহণ করতে পারে । কোন আউটপুট ভ্যালু রিটার্ন করতে পারে না ।

সুবিধা: -----------------

প্রসিডিউরাল প্রোগ্রামিং শেখা এবং বোঝা সহজ ।

প্রসিডিউরাল প্রোগ্রামগুলি দ্রুত রান হয় ।

অসুবিধা: -----------------

প্রসিডিউরাল কোড প্রায়শই রিউস করা কঠিন ।

বড় প্রজেক্টের ক্ষেত্রে প্রসিডিউরাল কোড বোঝা কঠিন হতে পারে ।

প্রসিডিউরাল প্রোগ্রামগুলি বড় এবং জটিল প্রজেক্টের জন্য স্কেল করা কঠিন হতে পারে ।

অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং কি ?

---------------------------------

অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং হলো এক ধরণের প্রোগ্রামিং প্যারাডাইম যেখানে মেথড এবং প্রপার্টিকে অবজেক্ট এর মধ্যে অর্গানাইজ করে প্রোগ্রাম লেখা হয় ।

সুবিধা: -----------------

কোড চেঞ্জ করা সহজ ।

কোড রিইউজ করা সহজ ।

নতুন ফিচারস যোগ করা সহজ ।

অসুবিধা: -----------------

OOP এর কন্সেপ্টস গুলো শেখা জটিল হতে পারে ।

OOP তে প্রোগ্রামিং করলে সাধারণত বেশি মেমরি ব্যবহার হয় ।

প্রসিডিউরাল প্রোগ্রামিং VS অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং ?

|  |  |
| --- | --- |
| প্রসিডিউরাল প্রোগ্রামিং | অবজেক্ট ওরিয়েন্টেড |
| প্রসিডিউরাল প্রোগ্রামিং এ প্রসিডিউর গুলো গ্লোবাললি এক্সেসিবল, ফলে সহজে এক্সেস করা যায় | অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং এ একটা ক্লাস এর আন্ডারে মেথড এবং প্রপার্টি গুলো একত্রিত করা থাকে, ফলে সহজে এক্সেস করা যায় না |
| ডাটা হাইড নিয়ে কাজ করা যায় না | ডাটা হাইড নিয়ে কাজ করা যায় |
| There is no access specifier in procedural programming. | bject-oriented programming has access specifiers like private, public, protected, etc. |
| function greet($name) {  $message = "Hello, $name!";  echo $message;  }  $name = "Alice";  greet($name); // Output: Hello, Alice!  // Since there are no access specifiers, the variable $message within greet() is accessible here as well.  echo $message; // This might cause unintended side effects if not careful. | class Person {  private $name; // Private property, accessible only within the class  public function \_\_construct($name) {  $this->name = $name;  }  public function greet() {  $message = "Hello, " . $this->name . "!";  echo $message;  }  }  $alice = new Person("Alice");  $alice->greet(); // Output: Hello, Alice!  // Since $name is private, it cannot be accessed directly here.  // echo $alice->name; // This would cause an error. |
| In procedural programming, overloading is not possible. | Overloading is possible in object-oriented programming. |
| In procedural programming, the function is more important than the data. | Object-oriented programming is used for designing large and complex programs. |

Classes ----

১) ক্লাস হল একটি টেম্পলেট বা বুলুপ্রিন্ট যা অবজেক্ট তৈরির জন্য ব্যবহৃত হয় ।

২) ক্লাস অবজেক্টের স্ট্রাকচার (মেথড এবং প্রপার্টি) এবং কাস্টম টাইপ নির্ধারণ করে অথবা

Class is a programmer-defined data type.

৩) ক্লাস হল একটা স্ট্রাকচার (মেথড এবং প্রপার্টি) যেখানে স্পেসিফিক টাইপের একটা অবজেক্টে ক্রিয়েট করি ।

অবজেক্ট -----

অবজেক্ট হল ক্লাসের ইনস্ট্যান্স, যা স্পেসিফিক মেথড এবং প্রপার্টি ধারণ করে ।

অবজেক্টকে প্রেসেন্ট করতে ৩ টি জিনিস লাগে: -----------------

১) স্টেট/ attribute: অবজেক্টের ডাটা বা প্রপার্টি কে রিপ্রেজেন্ট করে ।

২) বিহেভিয়ার: অবজেক্টের অ্যাকশন বা মেথডকে রিপ্রেজেন্ট করে ।

৩) আইডেন্টিটি: অবজেক্টেকে ইউনিক করে তোলে । আইডেন্টিটি অন্যান্য অবজেক্ট থেকে একটি নির্দিষ্ট অবজেক্ট কে আলাদা করতে ব্যবহৃত হয়।

class Person {

public $name; // State

public $age; // State

public function sayHello() { // Behaviour

echo "Hello, my name is " . $this->name . " and I am " . $this->age . " years old.";

}

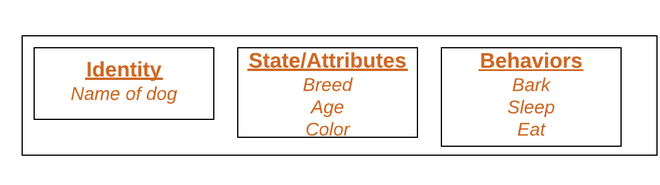
}

$person = new Person(); // Identity

$person->name = "John Doe";

$person->age = 30;

Object



অবজেক্ট ক্রিয়েট করার দুটি উপায়: -----------------

১) ক্লাস বেসড (অবজেক্ট বেসড): এটি পদ্ধতিটি জাভা, পিএইচপি এবং পাইথনের মতো ভাষায় ব্যবহৃত হয় ।

২) প্রোটোটাইপ বেসড: এই পদ্ধতিটি জাভাস্ক্রিপ্টের মতো ভাষায় ব্যবহৃত হয় ।

ক্লাস বেসড এবং প্রোটোটাইপ বেসড মধ্যে নির্বাচন করা: -----------------

১) ডেটা সিকিউরিটি এবং ইনহেরিটেন্স এর জন্য ক্লাস বেসড ।

২) ফ্লেক্সিবল এবং ডাইনামিক কোড এর জন্য প্রোটোটাইপ বেসড ।

ক্লাস প্রয়োজনীয়তা কেন ?

-----------------------------------

প্রোটোটাইপ বেসড প্রোগ্রামিং ক্ষেত্রে কোন নির্দিষ্ট স্ট্রাকচার ছাড়াই অবজেক্ট ক্রিয়েট করা যায় । অন্যদিকে, অবজেক্ট বেসড প্রোগ্রামিং ক্ষেত্রে অবজেক্ট তৈরি করার জন্য প্রথমে একটি স্ট্রাকচার নির্ধারণ করতে হয় যেখানে মেথড এবং প্রপার্টি গুলো থাকবে । এই স্ট্রাকচার কে ক্লাস বলা হয়, যার উপর বেস করে একটা অবজেক্টে ক্রিয়েট করা হয় ।

-----------------

মেথড ওভারলোডিং: যখন একটি ক্লাসে একই নামের একাধিক মেথড থাকে তবে প্যারামিটার অলাদা হয় তাকে মেথড ওভারলোডিং বলে । PHP মেথড ওভারলোডিং সমর্থন করে, তবে অন্যান্য ল্যাঙ্গুয়েজের মত ওভারলোডিং সমর্থন করে না (যেমন, java বা C++)। PHP-তে মেথড ওভারলোডিং করার জন্য ম্যাজিক মেথড \_\_call বা \_\_callStatic মেথড ব্যবহার করা হয় ।

class Calculator {

// Magic method for method overloading

public function \_\_call($name, $arguments) {

if ($name === 'multiply') {

if (count($arguments) === 2) {

return $this->multiplyTwo($arguments[0], $arguments[1]);

} elseif (count($arguments) === 3) {

return $this->multiplyThree($arguments[0], $arguments[1], $arguments[2]);

}

}

throw new \BadMethodCallException("Method '$name' not found.");

}

function multiplyTwo(){

}

function multiplyThree(){

}

}

$calculator = new Calculator();

echo $calculator->add(5, 3) . PHP\_EOL; // Output: 8

echo $calculator->multiply(2, 3) . PHP\_EOL; // Output: 6

echo $calculator->multiply(2, 3, 4) . PHP\_EOL; // Output: 24

২) মেথড ওভাররাইডিং: যখন প্যারেন্ট ক্লাস এবং চাইল্ড ক্লাসে এক্ই নামের মেথড থাকে যাদের প্যারামিটার(method signature) একই তখন তাকে মেথড ওভাররাইডিং বলে । এক্ষেত্ৰ চাইল্ড ক্লাসের মেথডেকে প্রায়রিটি বেশি । PHP মেথড ওভাররাইডিং সাপোর্ট করে

Method Overriding: -------------

class Animal {

public function makeSound() {

echo "Generic animal sound\n";

}

}

class Dog extends Animal {

public function makeSound() {

echo "Woof!\n";

}

}

class Cat extends Animal {

public function makeSound() {

echo "Meow!\n";

}

}

$dog = new Dog();

$dog->makeSound(); // Output: Woof!

$cat = new Cat();

$cat->makeSound(); // Output: Meow!