

# PHP for Base

Giảng viên: Bùi Quang Đăng

# Contents

1

Class and Object

2

Inheritance

3

Interface in PHP

4

Exercises

# PHP for Base

## Class and Object in PHP

# PHP for Base

## ❖ Class and Object

1

Lớp là tập hợp các đối tượng có cùng thuộc tính

2

Các phương thức, thuộc tính, biến

3

Đối tượng là tập hợp các thuộc tính mô tả đối tượng đó.

# PHP for Base

## ❖ Class and Object

- **Class**: Lớp là tập hợp các đối tượng có cùng thuộc tính giống nhau.

- **Cú pháp:**

```
class <Tên lớp>
```

```
{
```

```
    // Biến
```

```
    // Thuộc tính
```

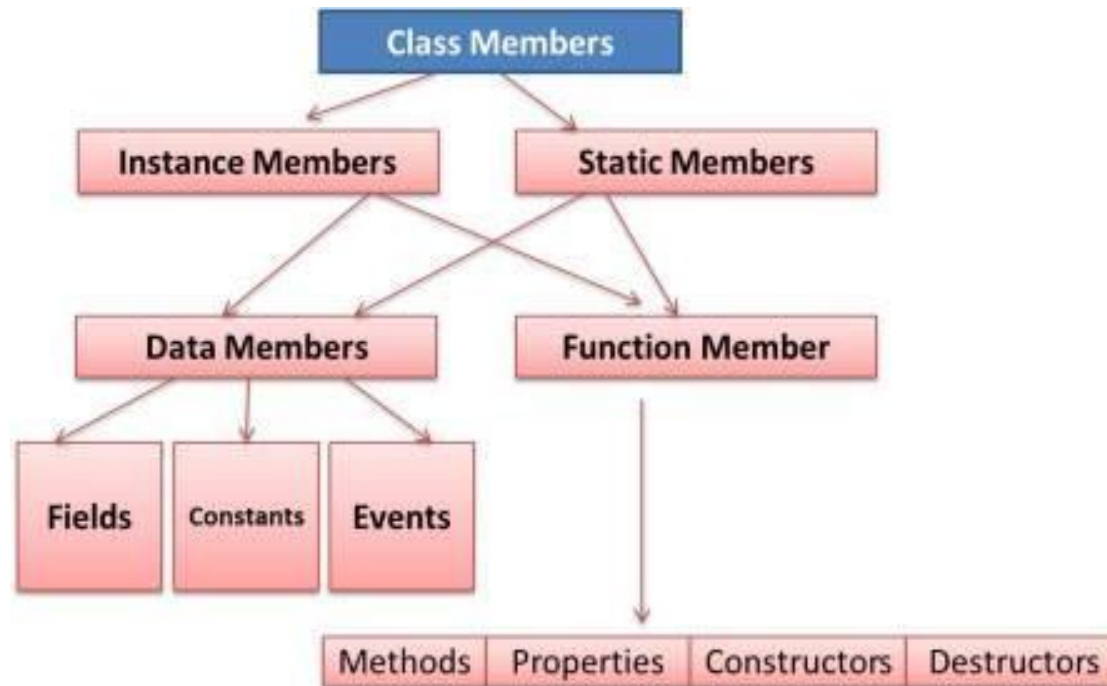
```
    // Phương thức
```

```
}
```

# PHP for Base

## ❖ Class and Object

- **Class**: Lớp là tập hợp các đối tượng có cùng thuộc tính giống nhau.



# PHP for Base

## ❖ Class and Object

- Ví dụ:

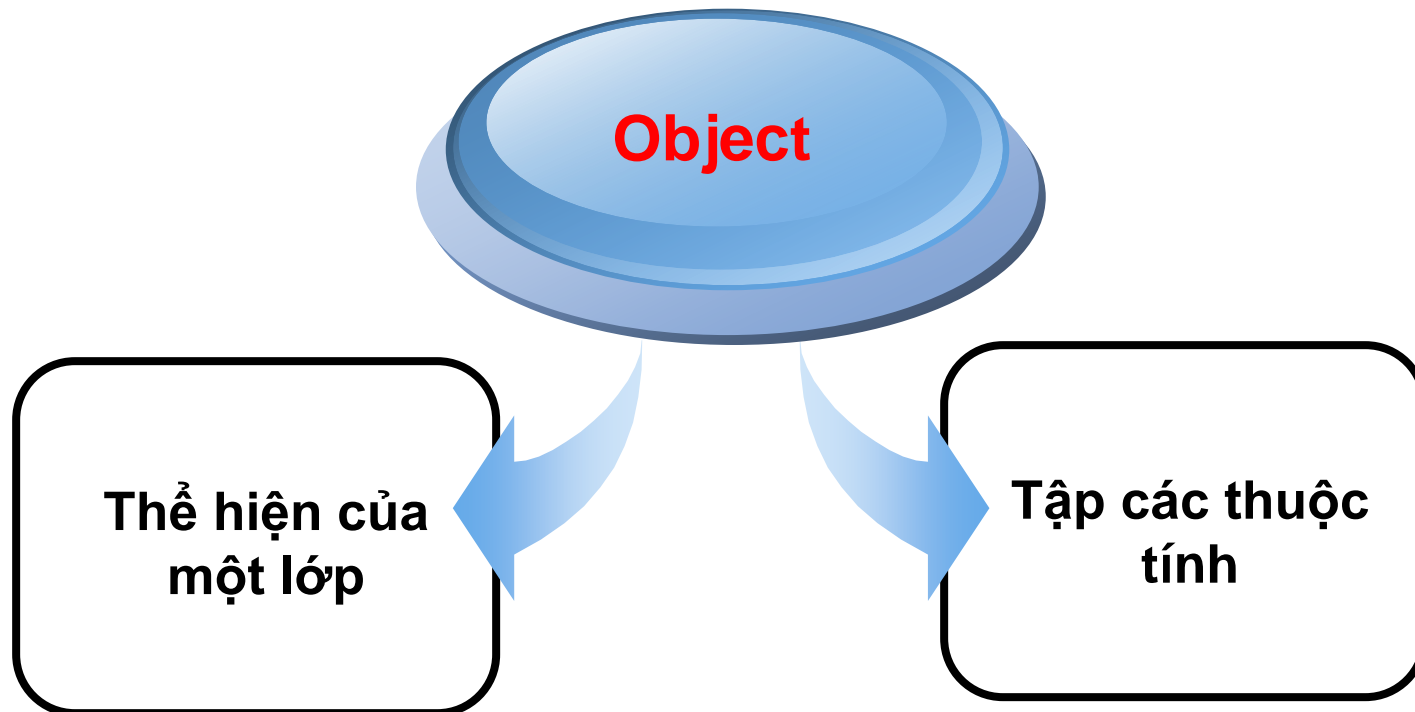
```
class SinhVien
{
    public $Id;
    public $MaSV;
    public $HoTen;
    public $GioiTinh;
    public $NgaySinh;
    public $DienThoai;
    public $Email;
    public $DiaChi;
    public function __construct($dc)
    {
        $this->DiaChi = $dc;
    }

    public function _destruct()
    {
    }
}
```

# PHP for Base

## ❖ Class and Object

- **Object:** Là một thể hiện cụ thể của lớp bao gồm tập hợp các thuộc tính mô tả đối tượng đó.





# PHP for Base

## ❖ Class and Object

- **Object:** Là một thể hiện cụ thể của lớp bao gồm tập hợp các thuộc tính mô tả đối tượng đó.
- Khai báo và tạo một đối tượng mới
  - [Đặt tên đối tượng] = **new** [Tên lớp]()
  - Sử dụng từ khóa new để khởi tạo một đối tượng mới.

**Ví dụ:**

```
$obj1 = new TinhToan();
```

```
$obj2 = new SinhVien();
```

```
$obj3 = new Person();
```

# PHP for Base

## ❖ Class and Object

- **function:** Là một tập các đoạn mã được viết để xử lý một số các công việc theo yêu cầu người lập trình.
- Khai báo và tạo một hàm trong PHP

**function** Ten\_ham(par1, par2,..., pn)

{

//Kết quả trả về nếu có sử dụng từ khóa return

}

# PHP for Base

## ❖ Class and Object

### ■ **function**

- Ví dụ:

```
function getFullName($pFullName)
{
    return $pFullName;
}
```

```
function TinhTong($a, $b)
{
    $tong = $a + $b;
}
```

# PHP for Base

## ❖ Class and Object

### ■ Function

- Ví dụ:

```
$tong = 0;
function TinhTong($a, $b)
{
    //global $tong;
    //Thay doi thong bien $tong sau loi goi ham
    $GLOBALS['tong'] = $a + $b;
}

TinhTong(3,5);

echo $tong;
```

- **Chú ý:** Sử dụng **\$GLOBALS** để xử lý gán giá trị cho biến toàn cục và chấp nhận giá trị của biến thay đổi sau lời gọi hàm

# PHP for Base

## ❖ Class and Object

### ■ **include:**

- Là hàm sử dụng trong PHP để khai báo để nhúng tài nguyên từ một trang php khác lên trang sử dụng.
- Khi không có file sẽ đưa ra cảnh báo
- Thường được sử dụng để nhúng các tài nguyên chung của trang và các chức năng hay sử dụng.

Ví dụ: `<?php     include("header.php")   ?>`

# PHP for Base

## ❖ Class and Object

### ■ **require:**

- Là hàm sử dụng trong PHP để khai báo để nhúng tài nguyên từ một trang php khác lên trang sử dụng.
- Khi không có file sẽ đưa ra lỗi, không thực hiện được nữa

Ví dụ: `<?php     require("header.php")   ?>`

# PHP for Base

## Inheritance

# PHP for Base

## ❖ Inheritance

- Là một lớp con được thừa kế các thuộc tính, phương thức từ lớp cha trong PHP.
- Cú pháp:

**class** Ten\_lop **extends** Ten\_lop\_co\_so

Ví dụ:

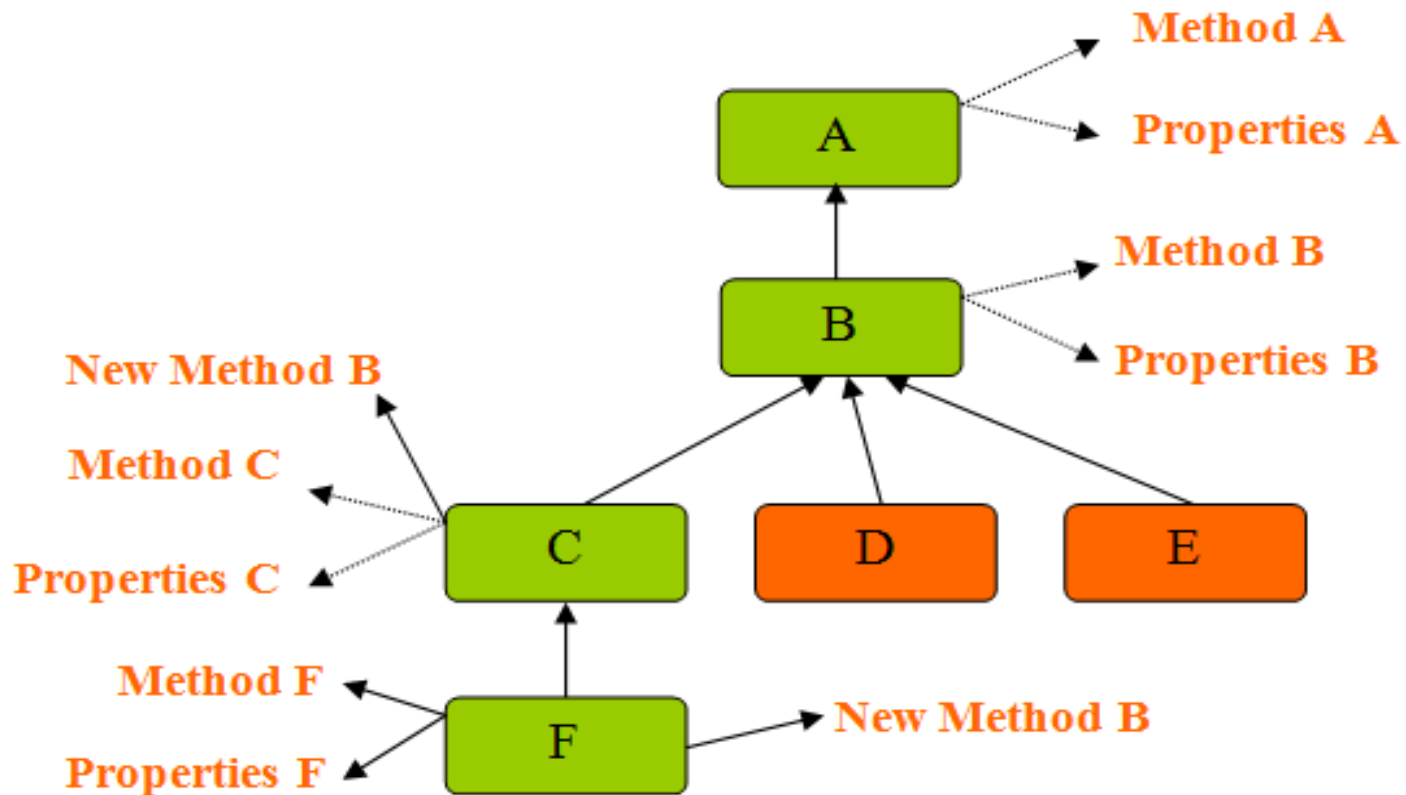
```
class Employee extends People
{
    //Biến, phương thức
}
```



# PHP for Base

## ❖ Inheritance

- Ví dụ:



# PHP for Base

## ❖ Inheritance

- Sử dụng hàm khởi tạo trong lớp kế thừa từ lớp cơ sở
- Ví dụ:

```
class MyOtherClass extends MyClass
{
    public function __construct()
    {
        parent::__construct(); // Call the parent class's constructor
        echo "A new constructor in " . __CLASS__ . "<br />";
    }

    public function newMethod()
    {
        echo "From a new method in " . __CLASS__ . "<br />";
    }
}
```

# PHP for Base

## ❖ Inheritance

Ví dụ: Cho một lớp animal có thông tin như sau:

```
class animal
{
    private $SoChan;
    private $SoMat;
    private $MauLong;

    public function SetSoChan($chan)
    {
        $this->SoChan = $chan;
    }

    public function GetSoChan()
    {
        return $this->SoChan;
    }

    public function TiengKeu($tiengkeu)
    {
        return $tiengkeu;
    }
}
```

```
class Dog extends animal
{
    private $Loai;

    public function SetLoai($loai)
    {
        $this->Loai = $loai;
    }

    public function GetLoai()
    {
        $this->Loai;
    }
}

//Khai bao va gan gia tri cho lop
$objDog = new Dog();
$objDog->SetSoChan(4);

echo $objDog->GetSoChan();
echo "<br/>";
echo $objDog->TiengKeu("Gâu gâu");
```

# PHP for Base

## ❖ Sử dụng từ khóa static

- Trong lập trình OOP có thể sử dụng từ khóa static để tạo ra thuộc tính, phương thức tĩnh.
- **Cú pháp:**

[Phạm vi truy cập] static Ten\_Bien

[Phạm vi truy cập] static [function] Ten\_Ham

Cách gọi: Ten\_lop :: Ten\_Ham hoặc Ten\_Bien

# PHP for Base

## ❖ Sử dụng từ khóa static

- Ví dụ:

```
class Dog extends animal
{
    .
    .
    .

    public static function Giong($sex)
    {
        return $sex;
    }
}

echo Dog::Giong("Giống đực");
```

# PHP for Base

## Interface

# PHP for Base

## ❖ Interface

- ✓ Là một lớp giao diện quy định các phương thức sẽ được sử dụng ở lớp thực hiện (**implement**)
- ✓ Nếu không khai báo đầy đủ các phương thức trong lớp thực hiện sẽ báo lỗi
- ✓ Sử dụng từ khóa **interface** để khai báo lớp giao diện, **các phương thức chỉ thực hiện khai báo định danh, không có nội dung xử lý.**
- ✓ Các phương thức đều sử dụng **public**

# PHP for Base

## ❖ Interface

✓ Cú pháp:

```
interface Ten_lop_giao_dien
```

```
{
```

```
    //Khai báo định danh các phương thức
```

```
}
```

```
class Ten_lop_su_dung implements Ten_lop_giao_dien
```

```
{
```

```
    //Khai báo và viết chi tiết nội dung xử lý của các phương thức
```

```
}
```



# PHP for Base

## ❖ Interface

✓ Ví dụ:

```
interface Action
{
    public function DanhSach();
    public function ThemMoi();
    public function CapNhatThongTin();
    public function XoaThongTin();
}
```

```
$data = new DataAccess();
```

```
echo $data->ThemMoi();
```

```
class DataAccess implements Action
{
    public function DanhSach()
    {
        //Xu ly lay danh sach
    }
    public function ThemMoi()
    {
        //Thuc hien xu ly them moi thong tin
        return "Thêm mới thông tin thành công";
    }

    public function CapNhatThongTin()
    {
        //Thuc hien xu ly cap nhat thong tin
    }

    public function XoaThongTin()
    {
        //Thuc hien xu ly xoa thong tin
    }
}
```



# Thank You !