

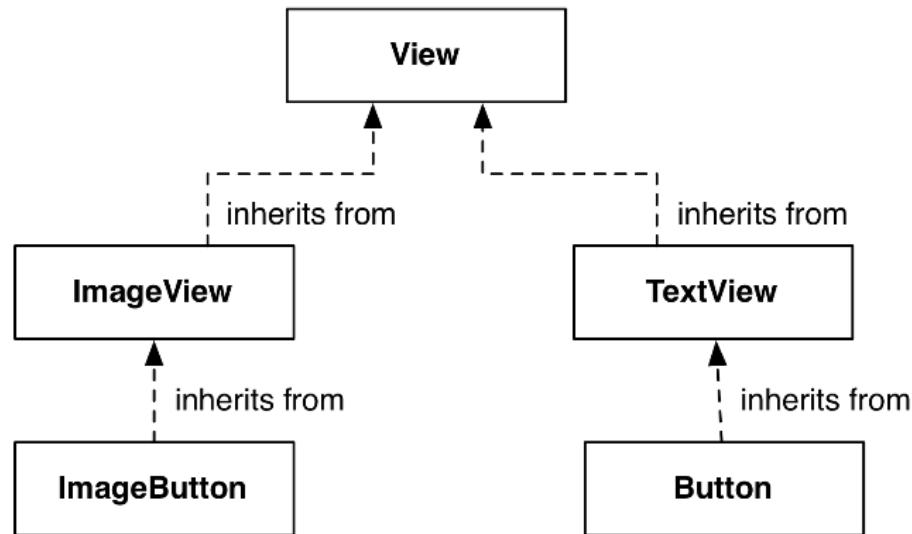


มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University

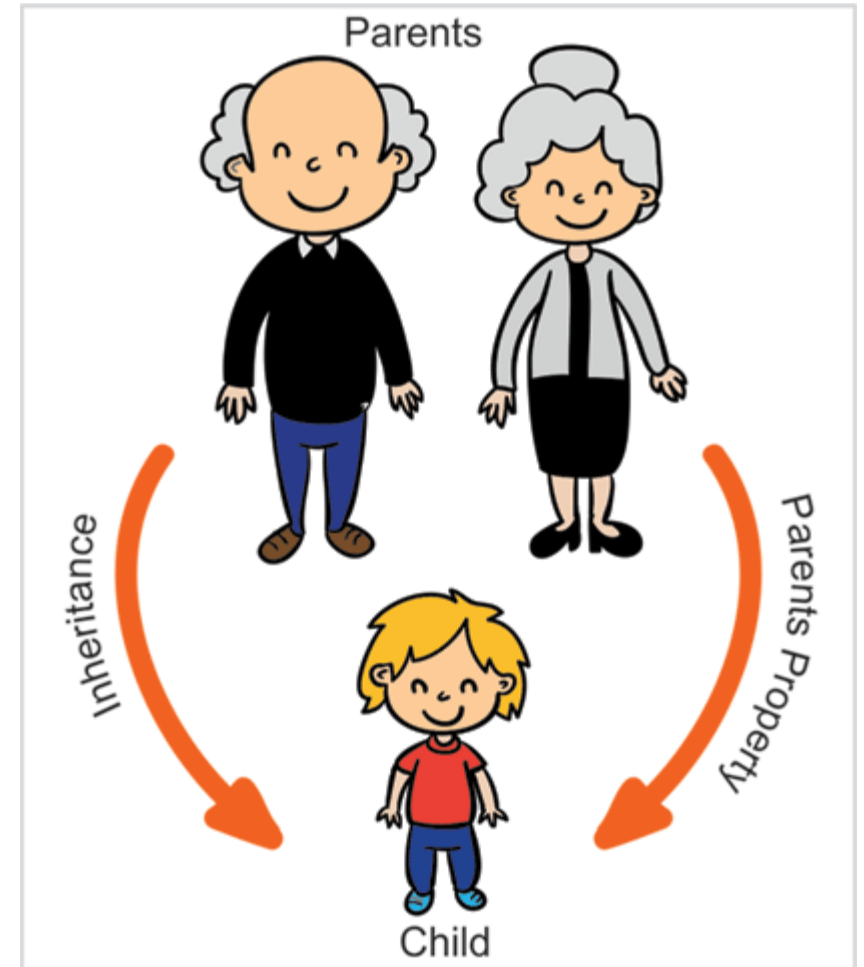
OOP Inheritance

ผู้ช่วยศาสตราจารย์สมเกียรติ ช่อเหมือน (tko@webmail.npru.ac.th)
สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาศาสตร์และเทคโนโลยี

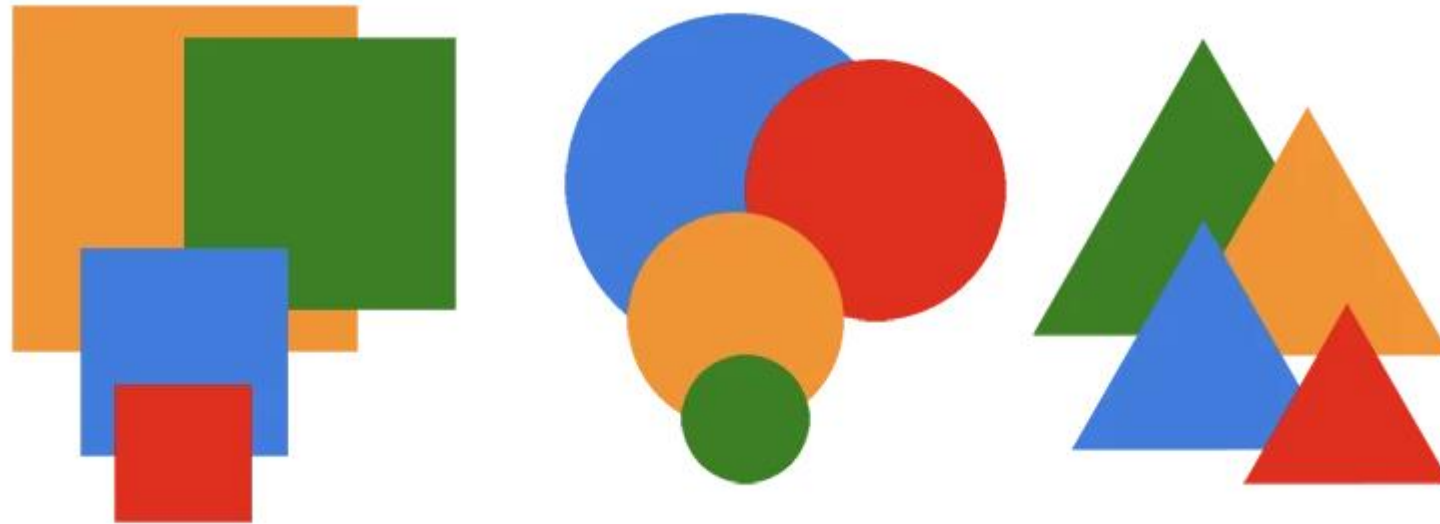
- Inheritance and Visibility Modifiers
- PHP Overriding Inherited Methods
- PHP final keyword
 - Prevent class inheritance and method overriding



- การสืบทอดคลาส (Inheritance)
- สร้างคลาสใหม่โดยสืบทอดจากคลาสเดิมได้
- การสืบทอดคุณสมบัติและพฤติกรรมจากคลาสหลัก
- เพิ่มเติมการทำงานหรือความสามารถเข้าไปได้
- นำโค้ดกลับมาใช้ใหม่ได้ (Reusable)



- กระบวนการการสืบทอด (Inheritance)
- การสร้าง Method ซ้ำใน Class ลูก
- Method ที่สร้างไว้แล้วที่ Class แม่
- กำหนดหน้าที่การทำงานใหม่ แตกต่างไปจากเดิม



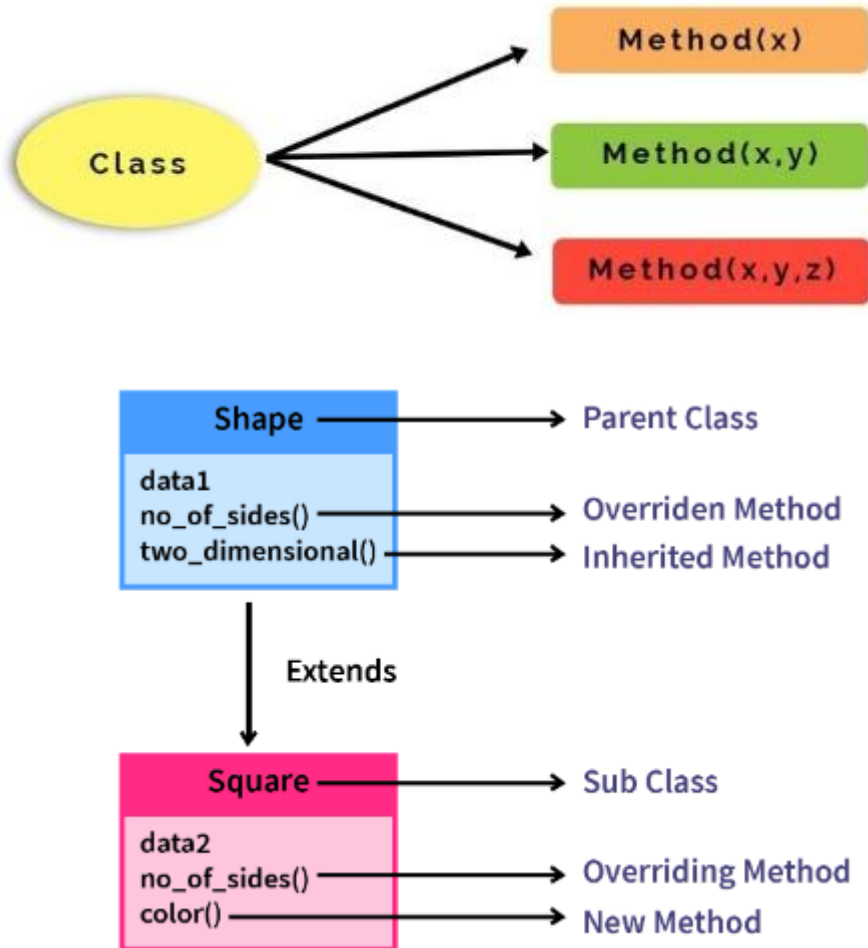
`protected` สามารถเรียกฟังก์ชันนั้นจาก `class` ที่สืบทอดได้



```
<?php
class ABC {
    private $abc = 'abc';
    protected function not_private() {
        return $abc;
    }
}

class DEF extends ABC {
    protected function call_me() {
        // when you call this function (which is on
        // the base class)
        // it will have no problems accessing $abc.
        $this->not_private();
    }
}
```

PHP Overriding Inherited Methods



```
<?php
class Human {
    protected $name;
    public function __construct($name){
        $this->name = $name;
    }
}

class Person extends Human {
    public $age;
    public function __construct($name, $age)
    {
        $this->name = $name;
        $this->age = $age;
    }
    public function getName(){
        return $this->name;
    }
}

$person = new Person('xyz', 18);
echo $person->getName(); # Prints: xyz
echo $person->age; # Prints: 18
```

parent constructor

- คลาสสืบทอดจากคลาสอื่น ใช้ extends ขยายคลาทย่อย
- constructor จากคลาสพื้นฐาน
- คีย์เวิร์ด parent ตัวดำเนินการ (::) ช่วยให้เรียก method ของคลาสพื้นฐาน



```
<?php
class Person extends Human {
    public $age;
    public function __construct($name, $age){
        $this->age = $age;
        parent::__construct($name); # calls the parent constructor
    }
    public function getName(){
        return $this->name;
    }
}
```

- final method ไม่ให้แทนที่เมธอดเดิม
- นำ version ของตัวเองไปใช้คลาสที่ extends
- คลาสที่สืบทอดยังคงสามารถเข้าถึงและเรียกเมธอดเหล่านี้ได้
- ไม่สามารถขยาย แก้ไข หรือนำไปใช้ใหม่

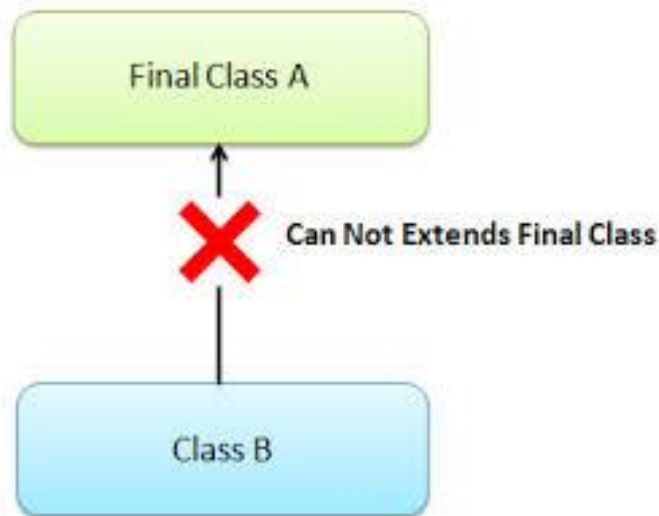


```
<?php
class Human {
    protected $name;
    final public function setName($name){
        $this->name = $name;
    }
}

class Person extends Human {
    public function setName($name){
        $this->name = $name;
    }
}

# Cannot override final method Human::setName()
```


- ประกาศคลาสเป็นขั้นสุดท้าย
- ไม่มีใครได้รับอนุญาตให้สืบทอดจากคลาสเหล่านี้
- ประกาศ final class เพื่อไม่ให้มีการสืบทอดจากคลาสใดได้อีก



```
<?php
final class Human {
    // ...
}

class Person extends Human {
    // ...
}

# Fatal error: Class Person may not inherit from final class (Human)
```

- การสืบทอดเป็นแนวคิดที่อนุญาตให้คลาสสืบทอดคุณสมบัติและพฤติกรรมจากคลาสอื่น
- คลาสที่ได้รับการสืบทอดมาเรียกว่าคลาสพื้นฐานหรือซูเปอร์คลาส
- คลาสที่สืบทอดมาเรียกว่าคลาสที่รับหรือคลาสย่อย

```
class Animal {
    public $name;

    public function eat() {
        echo "The animal is eating.";
    }
}
```

```
class Dog extends Animal {
    public function bark() {
        echo "The dog is barking.";
    }
}
```

```
// Creating an instance of the Dog class
$dog = new Dog();
$dog->name = "Max";
$dog->eat(); // Inherited method from the Animal class
$dog->bark(); // Method defined in the Dog class
```