1. Trong kế thừa, class con có phương thức muốn kế thừa phương thức của class cha ta dungf hàm parent::phương thức của class cha

VD: class cha{

Protected function phuongthuccha(){}

}

Class con{

Public function phuongthuccon(){

Parent::phuongthuccha();

}

}

1. Static method 😊 muốn gọi phương thức static ta dung cú pháp: Tenclass::tenphuongthuc();

VD: class hocsinh{

Public static function setTen(){}

}

->gọi: tạo đối tượng gọi cũng đc mà t dung static nên gọi theo cách của static

-> Gọi kiểu static: hocsinh::setTen();

1. Static properties(thuộc tính static)

Để sử dụng thuộc tính static kiểu private static $count =0;

Ta dung hàm self::$count;

Trả về kết quả cũng dung hàm self::$count đó

1. Abstract và interface

Abstract: đơn kế thừa – dung hàm extends để kế thừa

Interface: đa kế thừa – dung hàm imploments đề kế thừa

( đa kế thừa nghĩa là 1 class có thể kế thừa nhiều interface )

1 class có thể vừa kế thừa 1 abstract và kế thừa nhiều interface cùng lúc

khi khai báo interface hay abstract thì class kế thừa phải có các phương thức của interface hay abstract đó dù có dung hay không.

1. Đề lấy tên class đang sử dụng ta dung hàm get\_class(“ten class”);
2. Đề kiểm tra class có tồn tại hay không ta dung hàm class\_exists($đối tượng class);
3. Dung hàm \_\_autoload($class) để gọi tự động require\_once tất cả class 1 lần(nhớ là tên class phải trung tên với tên của file)

VD: function \_\_autoload($classname){

Require\_once($classname.”.php”);

}