

LOCALIZATION OF E-GOVERNANCE PROJECT

PHP 5 Classes and Objects

Prepared by:

Project Team,

The e-platform model,

DIT, MoIC, RGoB

Contents

1. Introduction	2
2. Sample Exercise (Class)	2
3. Using Constructors	5
4. Using private	6
5. Inheritance Concept	8
6. Using final	10
7. Using Destructors	11
8. get_class() and get_parent_class()	12
9. Autoloading Objects	14
10. Visibility	15
11. Scope Resolution Operator (::)	19
12. Static Keyword	21
13. Class Abstraction	22
14. Object Iteration	25
15. Magic Methods	26
16. Patterns	27
17. Late Static Bindings	30
18. Objects and references	34
19. Interfaces	35
20. Exceptions	37

Introduction

In PHP, a class is simply a set of program statements which perform a specific task. A typical class definition contains both variables and functions, and serves as the template from which to spawn specific instances of that class.

These specific instances of a class are referred to as objects. Every object has certain characteristics, or properties, and certain pre-defined functions, or methods. These properties and methods of the object correspond directly with the variables and functions within the class definition.

Once a class has been defined, PHP allows you to spawn as many instances of the class as you like. Each of these instances is a completely independent object, with its own properties and methods, and can therefore be manipulated independently of other objects. This comes in handy in situations where you need to spawn more than one instance of an object - for example, two simultaneous database links for two simultaneous queries, or two shopping carts.

Classes also help you keep your code modular - you can define a class in a separate file, and include that file only in the scripts where you plan to use the class - and simplify code changes, since you only need to edit a single file to add new functionality to all your spawned objects.

Sample Exercise (Class)

To understand this better, pick an animal, any animal. For example, we can take bear. every bear has certain characteristics - age, weight, sex - which are equivalent to object properties. And every bear can perform certain activities - eat, sleep, walk, run, mate - all of which are equivalent to object methods.

Let's take it a little further. Since all bears share certain characteristics, it is possible to conceive of a template `Bear()`, which defines the basic characteristics and abilities of every bear on the planet. Once this `Bear()` ("class") is used to create a new `$bear` ("object"), the individual characteristics of the newly-created `Bear` can be manipulated independently of other `Bears` that may be created from the template.

Now, if you sat down to code this class in PHP 5, it would probably look something like this:

```

<?php
    // PHP 5 // class definition
    class Bear {
        // define properties
        public $name;
        public $weight;
        public $age;
        public $sex;
        public $colour;

        // define methods
        public function eat() {
            echo $this->name." is eating...\n";
        }
        public function run() {
            echo $this->name." is running...\n";
        }
        public function kill() {
            echo $this->name." is killing prey...\n";
        }
        public function sleep() {
            echo $this->name." is sleeping...\n";
        }
    }
?>
<?php
    // my first bear
    $daddy = new Bear;
    // give him a name
    $daddy->name = "Daddy Bear";
    // how old is he
    $daddy->age = 8;
    // what sex is he
    $daddy->sex = "male";
    // what colour is his coat
    $daddy->colour = "black";
    // how much does he weigh
    $daddy->weight = 300;

```

```
// give daddy a wife
$mommy = new Bear;
$mommy->name = "Mommy Bear";
$mommy->age = 7;
$mommy->sex = "female";
$mommy->colour = "black";
$mommy->weight = 310;

// and a baby to complete the family
$baby = new Bear;
$baby->name = "Baby Bear";
$baby->age = 1;
$baby->sex = "male";
$baby->colour = "black";
$baby->weight = 180;

print("<h2>PHP Class and Object Example 1</h1>");
// a nice evening in the Bear family
// daddy kills prey and brings it home
$daddy->kill();
echo "<br/>";
// mommy eats it
$mommy->eat();
echo "<br/>";
// and so does baby
$baby->eat();
echo "<br/>";
// mommy sleeps
$mommy->sleep();
echo "<br/>";
// and so does daddy
$daddy->sleep();
echo "<br/>";
// baby eats some more
$baby->eat();
echo "<br/>";
```

?>

[Click here to download full PDF material](#)