

OO Upgrade: PHP 5's Object Model

*Greg Ferguson
May 6, 2005*

ICS 139W

Abstract

The creators of PHP did not intend for PHP to contain object oriented features. As PHP became popular programmers began to demand the inclusion of object oriented features in PHP. PHP 5 is Zend Technologies' solution to providing a strong object model in PHP. The changes to the programming language include the addition of class security, passing objects by reference, introduction of predefined methods, abstract class features, interfaces, and exception handling. The addition of these object oriented features allows PHP to become a programming language capable of building enterprise class programs.

1 Introduction

Popular programming languages today are designed with object oriented features. Object oriented programming is popular among the programming community today because it provides a structure to a program and allows for abstraction. PHP versions prior to version 5 did not include many object oriented features that are essential to object oriented programming. Pressure from the programming community led to an upgrade of PHP's object oriented features. PHP version 5 includes fixes as well as new object model which make it a better programming language. Improvements to the language include:

- Passing objects by reference by default.
- Introduction of predefined methods.
- Introduction of abstract classes and interfaces.
- Introduction of exception handling.

2 Class Security

The creators of PHP decided to include class security for the first time in PHP 5. Programmers can define class variables and methods as public, protected, or private. Giving programmers the ability to use class security allows the programmer to ensure which classes have access to what data.

3 Passing Objects by Reference

One of the most important fixes in the newest version of PHP is improved object handling. In previous versions of PHP when an object was passed as an argument to a function a copy of that object was created for use within the function. A programmer could specify that they wanted the variable to be passed by reference but it wasn't the default. In version 5, PHP passes a reference to an object and not a copy of the object as a parameter. Consider code segment 1.

- Introduction of class security.

```

class foo
{
    function foobar ($barObj)
    {
        $barObj->temp =
            "set by foo::foobar";
    }
}
class bar
{
    var $temp = "default value";
}

$a = new foo();
$b = new bar();
$a->foobar($b);
echo $b->temp;

```

Code Segment 1: *Example of passing an object by reference.*

The output of this code segment is different depending on the version of PHP you are using. PHP 4 outputs “default value” because objects are passed as a copy of the original object and any changes to the copy will not affect the original object. PHP 5 outputs “set by foo:bar” because objects are passed by reference in PHP 5. Passing objects by reference allows a more intuitive approach to programming and allows the programmer to have control over when an object is copied [4].

4 Predefined Methods

The designers of PHP included several predefined methods.

- `__clone()`
- `__construct()` and `__destruct()`
- `__get()` and `__set()`
- `__call()`

The inclusion of these predefined methods gives programmers added guarantees; programmers will know more about how other classes are written [5],[6].

4.1 The `__clone()` Method

The `__clone()` method gives the programmer control over how objects are copied. The `__clone()` method allows the programmer to decide what aspects of an object are copied to the new object when a cloned object is created. If the programmer chooses not to include a clone method inside of their class a default clone method will be used. The default clone method copies the object bit for bit.

4.2 The `__construct()` and `__destruct()` Methods

The `__construct()` and `__destruct()` methods were as a result of a design decision to have unified names for the constructor and destructor. The design decision was made to ensure that all child classes know exactly how to access a parent’s constructor. This feature is helpful in the case that your class is moved under a new parent because your code will not have to change [4]. A parent’s constructor can always be called by the line, `parent::__construct();`.

4.3 The `__get()` and `__set()` Methods

Programmers often access class variables from outside of the class. The `__get()` and `__set()` methods were created specifically to aid in the access of class variables.. The `__get()` method takes a variable name as an argument. And the `__set()` method takes a variable

```

class foo {
    public __construct() {
        //no arguments
    }
    public __call($methName,$methArgs) {
        if ($methodName == "__construct"){
            //one or more arguments
        }
    }
}

```

Code Segment 2 *Example of an overloaded constructor using __call() method*

name and a new value as arguments.

4.4 The __call() Method

The __call() method was created to assist programmers with method overloading. PHP does not allow method overloading. When a method is called and it does not exist within the class the __call() method within that class is executed. The __call() method is provided the name of the method that was called and the arguments that were provided to the non-existent method. Knowing these two facts the programmer uses the __call() method to decide what should happen in the program. Code segment 2 shows an overloaded constructor: using the __call() method

5 Abstract Classes and Interfaces

The developers of PHP included abstract classes and interfaces to help programmers. Abstract classes and inheritance help programmers to ensure classes will contain methods of the programmers choosing. Although PHP does not include multiple inheritance, it does allow abstract classes to be used in combination with interfaces. This

combination provides similar functionality to multiple inheritance.

6 Exception Handling

When building an enterprise level application it becomes necessary to ensure a program will not run into a fatal error which will cause the program to stop execution. To provide this functionality exception handling is included in the newest version of PHP. Exceptions can be caught and thrown by the programmer to ensure correct execution of the program.

7 Conclusion

This version of PHP is not perfect; improvements can be made to PHP in order to make PHP more like a modern programming language. All versions of PHP have not contained method overloading. A programmer can not define multiple methods which share the same name but take different arguments. The inclusion of method overloading would allow for improved coding practices. In section 4.4 I mentioned that the __call() method could be used to provide overloading but this is not a good solution when more than one method is being overloaded.. Another problem with PHP is that it does not include multiple inheritance. The designers of PHP took the same stance as the designers of Java; they decided to leave multiple inheritance out of their programming language. By leaving this feature out of their programming language they do not have to deal with the problems that arise from

multiple inheritance; but they leave out a feature that may be the best solution to a problem.

Predefined methods and class security along with the decision to pass objects by reference, support the inclusion of object oriented features class abstraction, interfaces, and exception handling. The new features of the PHP programming language were created with the programmer in mind. These features enhance the object model of PHP and make PHP a better programming language.

References

- [1] Braam, Paul A.J. Design Patterns applied to Web Programming in PHP. University of Amsterdam. 14 Marth 2004.
- [2] Brenda, Jan et al. Phalanger: Compiling PHP Applications for the Microsoft .NET Platform. Charles University in Prague.
- [3] Leo, Cedric. Evaluation of Dynamic Reconfigurable WWW-content Prototypes. BEng Honours Multimedia Systems. May 2002.
- [4] Zend Technologies. Zend Engine Version 2.0 Feature Overview and Design.
- [5] Zend Technologies. *Magic Methods*. <http://www.zend.com/manual/language.oop5.magic.php>
- [6] Zend Technologies. *Overloading*. <http://www.zend.com/manual/language.oop5.overloading.php>