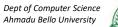


WEB APPLICATION ENGINEERING II

Lecture #9

Umar Ibrahim Enesi





Objectives

- Gain understanding of various constructs used in Object Oriented PHP.
- Understand the advantages of using OOP style over procedural style of programming.
- Experiment with PHP DateTime API.
- Have an introduction on SPL library.

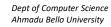
Dept of Computer Science Ahmadu Bello University



Introduction to Object Oriented PHP

- So far, most of the example scripts have been written with PHP in a procedural approach.
- OOP lay emphases on classes and objects.
- PHP provides mechanisms that makes it possible and convenient to use the object oriented style.

06-Nov-16 Umar Ibrahim Enesi





PHP OOP Concepts – Classes

A valid class name starts with a letter or underscore, followed by any number of letters, numbers, or underscores

- Classes are used to define objects.
- With a class definition, any number of objects can be created from it.

```
class class_name{
    /*properties and method definitions*/
}
```

- class_name can be any valid label, provided it is not a PHP reserved word eg. Student, Shape, User
- Constants and variable definitions make up the properties while...
- Functions make up the method definitions



Properties

- Constants and variables constitutes class properties.
- Class properties may be static (class) or non-static (instance)
- Use static keyword to declare a static property.
- Static properties are accessible without needing an instantiation of the class.
- Non-static properties are only accessible through an instance of the class.

06-Nov-16 Umar Ibrahim Enesi



Methods Definitions

- Functions constitutes class methods.
- Class methods may be static or non-static (instance)
- Use static keyword to declare a static method.
- Static methods are accessible without needing an instantiation of the class.
- Non-static methods are only accessible through an instance of the class.

Dept of Computer Science Ahmadu Bello University



• To access static members (properties and methods), use :: (double colon) operator.

```
Class name Operator Static Method

Student::getStudentCount();
```

• To access non-static members, use -> operator.

```
Object variable Operator Instance method $std_obj->getName();
```

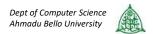
06-Nov-16 Umar Ibrahim Enesi

Dept of Computer Science Ahmadu Bello University



Accessing non-static members and \$this Variable

- Non-static properties and methods cannot be referenced from static methods (context).
- But static members can be accessed in non-static context.
- \$this variable references the current object in view.
- \$this is always made available within non-static methods.
- \$this is also available in static methods called from within an object context.



Constructors and Destructors

- construct(...)
- Automatically called when an object is created.
 - destruct()
- Automatically called before an object is destroyed

06-Nov-16 Umar Ibrahim Enesi



Visibility Modifiers

- Specifies the scope in which class members are accessible.
 - public : Variable is visible both within and outside the class it is declared (everywhere)
 - private : Variable is visible only within the class it is declared
 - protected : Variable is visible in the class it is declared and in subclasses of the class

```
private $x = "...";
public function getName() { ... }
```



Inheritance

- A class (subclass) can inherit from another class (super class) using the extends keyword.
- The subclass inherits all of the public and protected methods from the parent class.
- A subclass may override methods inherited from its ancestors.
- A subclass may include additional methods and properties to the ones inherited from its ancestors.

06-Nov-16 Umar Ibrahim Enesi 1



Interface

- Interfaces formalises the creation of classes and its related object.
- Interfaces may contain public abstract methods and constants.
- A class implements an interface by using the implements keyword.
- A class may implement as many interface as it wishes.

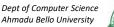
```
public interface x{
     public function a(...);
     ...
}
public class myclass implements x{
     public function a(...){...}
}
```



Comparing Objects

- Use "==" to compare objects by type, properties and values.
 - Two objects are equal if:
 - They are instance of the same class,
 - · Have the same set of properties, and
 - Have the same set of corresponding values.
- Use "===" to compare objects by reference
 - Two objects are equal if and only if their reference is the same.

06-Nov-16 Umar Ibrahim Enesi 1





Benefits of Employing Object Oriented PHP

- Code Re-Use
- Easy Debugging and Testing
- Code Organization and Maintenance
- Library Integration
- Use of PHP Framework

Dept of Computer Science Ahmadu Bello University



Procedural PHP vs Object Oriented PHP

Zoo App.

```
...
$animals= array("Lion", "Snake", ..., "Monkey",...);
foreach($animals as $animal){
    if($animal =="Lion")
        echo "roar...";
    else if ($animal =="Snake")
        echo "hisss...";
    ...
}

?>
```

What if we have more than 100 animals?

06-Nov-16 Umar Ibrahim Enesi 15

Dept of Computer Science Ahmadu Bello University



Procedural PHP vs Object Oriented PHP

Zoo App.

```
<?php

...
   //all form of animals are subclass of Animal Class
   $animals= array(new Lion(), new Snake(), ..., new Monkey(),...);
   foreach($animals as $animal){
        $animal->makeNoise();
}
```



Standard PHP Library (SPL)

- The Standard PHP Library (SPL) is a collection of interfaces and classes that are meant to solve common problems.
- There is more control over iteration of object properties if an object is created from SPL classes.
- Many of the classes defined in SPL implements various data structures like Queue, List, Stacks e.t.c.
- Many vendor libraries also rely on SPL.

06-Nov-16 Umar Ibrahim Enesi 1

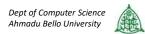


Example: PHP DateTime Object

- DateTime Object is used to manage date and time conveniently in an object oriented fashion.
- PHP provides various classes for managing Date and Time.
 - DateTime class
 - DateTimeImmutable class
 - DateInterval class
 - DatePeriod class

06-Nov-16 Umar Ibrahim Enesi 18

9

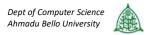


DateTime and DateInterval Classes

```
/*create a date object that represents current date
and time*/
$dt = new DateTime();

/*create a date interval object that represents
and interval of 2 days, 1 hour and 5 minutes*/
$dt_intvl = new DateInterval("P2DT1H5M");
```

06-Nov-16 Umar Ibrahim Enesi 19



DateTime and DateInterval Classes

```
/*add an interval to a date*/
$dt -> add($dt_intvl);

/*returns a string representation of the date in a specific format (eg 10th August, 2010) */
$dt_str = $dt->format("dS F, Y");
echo $dt_str;
```



Key Points

- PHP provides mechanisms that makes it possible to write programs in an object oriented style.
 - Class declaration with class keyword
 - Method and properties declaration
 - · Static and non-static members
 - Visibility modifiers: public, protected, private
 - Interface and Inheritance support
 - Member reference operator: "->" and "::"
 - Type sniffing with instanceof

06-Nov-16 Umar Ibrahim Enesi 2:



Reference

- PHP Documentation Manual
- Object-Oriented Programming with PHP5 by Hasin Hayder
- Murach PHP and MySQL
- http://www.phpclasses.org/blog/post/178-Why-is-it-better-to-develop-in-PHP-with-classes-OOP.html
- http://www.htmlgoodies.com/beyond/php/article.php/3909681