Object Oriented Programming with PHP (2)

Topics

- Final Keyword
- Class Abstraction
- Object Interfaces

Final Keyword

- PHP introduces “Final” keyword, that prevent overriding of sub-class method
- “Final” can be implement at method and class
- Final class means that class cannot be extended

Final Keyword (cont.)

```
<?php
class BaseClass {
   public function test() {
       echo "BaseClass::test() called\n";
   }

   final public function moreTesting() {
       echo "BaseClass::moreTesting() called\n";
   }
}

class ChildClass extends BaseClass {
   public function moreTesting() {
       echo "ChildClass::moreTesting() called\n";
   }
}

// Results in Fatal error: Cannot override final method BaseClass
::moreTesting()
?>
```

Final Keyword (cont.)

```
<?php
final class BaseClass {
   public function test() {
       echo "BaseClass::test() called\n";
   }

   // Here it doesn't matter if you specify the function as final
 or not
   final public function moreTesting() {
       echo "BaseClass::moreTesting() called\n";
   }
}

class ChildClass extends BaseClass {
}
// Results in Fatal error: Class ChildClass may not inherit from
final class (BaseClass)
?>
```

Class Abstraction

- Class that is defined as an abstract cannot be instantiated
- Class that consist of one abstract method, must be declared as abstract class

Class Abstraction

```
abstract class AbstractClass
{
    // Force Extending class to define this method
    abstract protected function getValue();
    abstract protected function prefixValue($prefix);

    // Common method
    public function printOut() {
        print $this->getValue() . "\n";
    }
}

class ConcreteClass1 extends AbstractClass
{
    protected function getValue() {
        return "ConcreteClass1";
    }

    public function prefixValue($prefix) {
        return "{$prefix}ConcreteClass1";
    }
}
```

Object Interfaces

- Object Interfaces allow us to define method without detail
- Methods of Interfaces must be define with public modifier
- Using "implement" keyword to use object interfaces

Object Interfaces (cont.)

- Class cannot implement more than one class that consist of the same method
- Interface can be inherited as same as usual class using “extend” keyword
- Class that implement the interfaces must be use all methods it owned, with the same specification

Object Interfaces (cont.)

```php
<?php
interface a
{
    public function foo();
}

interface b extends a
{
    public function baz(Baz $baz);
}

// This will work
class c implements b
{
    public function foo()
    {
    }

    public function baz(Baz $baz)
    {
    }
}
?>
```

Tugas

- Monyet merupakan salah satu dari jenis Hewan, Monyet mempunyai kemampuan untuk Berjalan, Makan dan Bersuara, tapi suatu saat Monyet apabila dilatih akan bisa Berkendara seperti naik sepeda yang tidak dipunyai oleh hewan lainnya.


```
<?php
        echo “Terima Kasih….!!!”
?>
```