

PHP CLASSES AND OBJECTS

Andrew Sheehan

Boston University
Metropolitan College

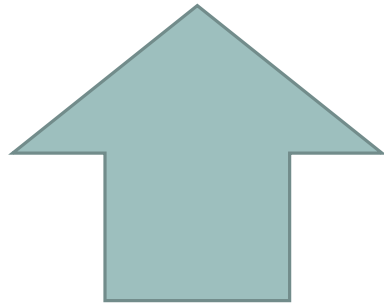
CLASS DEFINITION

```
<?php  
class Automobile {  
    /* class body */  
}  
?>
```

- ❖ Class names cannot be reserved word (part of the language)
- ❖ Cannot start with numbers

CLASS
'NEW'

```
$auto = new Automobile();
```



To create an instance, `new` must be used.

CLASS

\$THIS

```
$this->firstName;  
$this->displayName();
```

Use `$this->` within an object

CLASS CONSTRUCTOR

```
__constructor() {...}
```

For initialization before use.

CLASS CONSTRUCTOR

```
<?php
```

```
class Person {
```

```
    private $firstname;
```

```
    private $middleName;
```

```
    private $lastname;
```

```
    __constructor($first = " ", $middle = " ", $last = " ") {
```

```
        $this->$firstname = $first;
```

```
        $this->$middleName = $middle;
```

```
        $this->$lastname = $last;
```

```
    }
```

```
?>
```

Default function argument
values




CLASS

PARENT CONSTRUCTOR

```
<?php
class BaseClass {
    function __construct() {
        print "In BaseClass constructor\n";
    }
}

class SubClass extends BaseClass {
    function __construct() {
        parent::__construct();
        print "In SubClass constructor\n";
    }
}
?>
```



CLASS VISIBILITY

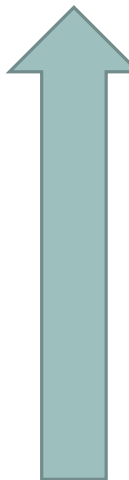
var
or


private

public

protected


class itself


Seen anywhere


class & subclasses

CLASS VISIBILITY

```
private $socialNumber = null;
```

private properties can not be
accessed (seen or used) in subclasses

CLASS VISIBILITY

```
private $socialNumber = null;
```

Create **protected methods** in the parent class allow subclasses to touch private attributes.

CLASS

SCOPE RESOLUTION OPERATOR

```
<?php
```

```
class MyClass {
```

```
    const CONST_VALUE = 'A constant value';
```

```
}
```

```
echo MyClass::CONST_VALUE;
```

```
?>
```

CLASS FINAL

```
<?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()
?>
```

CLASS STATIC FUNCTION

```
<?php
class Foo {
    public static function aStaticMethod() {
        // ...
    }
}

Foo::aStaticMethod();
$classname = 'Foo';
$classname::aStaticMethod(); // As of PHP 5.3.0
?>
```

CLASS STATIC ATTRIBUTE

```
<?php
class Foo
{
    public static $my_static = 'foo';

    public function staticValue() {
        return self::$my_static;
    }
}

class Bar extends Foo
{
    public function fooStatic() {
        return parent::$my_static;
    }
}

?>
```

CLASS OVERRIDING

```
class SomeOtherClass extends SomeClass{  
    function scream($count = 1) {  
        for ($i = 0; $i < $count; $i++) {  
            echo 'Whooohoo!<br>';  
        }  
    }  
}
```

```
class SomeClass {  
    function scream($count = 1) {  
        for ($i = 0; $i < $count; $i++) {  
            echo 'Eek!<br>';  
        }  
    }  
}
```