Inheritance

Inheritance is the means by which one or more classes can be derived from a base class.

A class that inherits from another is said to be a subclass of it. This relationship is often described in terms of parents and children. A child class is derived from and inherits characteristics from the parent. These characteristics consist of both properties and methods. The child class will typically add new functionality to that provided by its parent (also known as a superclass); for this reason, a child class is said to extend its parent.

Working with Inheritance

The first step in building an inheritance tree is to find the elements of the base class that don't fit together or that need to be handled differently.

We can think of an easy example being with a class for Dog and a class for Cat. Neither animal would share all functions or properties, so it would be weird to create one class for Dog and then put properties and methods for a Cat object. Instead, we can create a class that both classes can extend from and have unique methods to themselves.

Example:

```
class Animal {
	public $firstName;
	public $lastName;
	public $scientificName;
	public $gender;
	public $weight;
	public $bark;
	public $meow;

	function __construct($scientificName, $firstName, $lastName, $gender, $weight, $bark, $meow) {
		$this->scientificName = $scientificName;
		$this->firstName = $firstName;
		$this->lastName = $lastName;
		$this->gender = $gender;
		$this->weight = $weight;
		$this->bark = $bark;
		$this->meow = $meow;
	}

	function getName() {
	return “This is my “ . $this->firstName .
		“ and last ” . $this->lastName;
	}
}
class Cat extends Animal {
	function greet() {
	return $this->meow”;
	}
```

```
}

class Dog extends Animal {
        function hello() {
        return $this->bark;
        }
}

$cat = new Cat(“Cat”, “Mellow”, “Yellow”, “male”, 15);
print “Animal 1 is a ” . $cat->getName();
```

To create a child class, you must use the extends keyword in the class declaration. In the example, I created two new classes, Cat and Dog. Both extend the Animal class.

Because the derived classes do not define constructors, the parent class’s constructor is automatically invoked when they are instantiated. The child classes inherit access to all the parent’s public and protected methods (though not to private methods or properties). This means that you can call the getName() method on an object instantiated from the Cat class, even though getName() is defined in the Animal class: