Creating a Constructor Method

A constructor method is invoked when an object is created. You can use it to set things up, ensuring that essential properties are set and any necessary preliminary work is completed. Note that the method name begins with two underscore characters.

```
class Dog {
public $firstName;
public $lastName;
public $breed;

        function __construct($title, $firstName, $lastName, $breed) {
        $this->firstName = $firstName;
        $this->lastName = $lastName;
        $this->breed = $breed;
        }

        function getName() {
        return “{$this->firstName}” .
        “{$this->lastName}”;
        }
}
```

Once again, I gather functionality into the class, saving effort and duplication in the code that uses it. The __construct() method is invoked when an object is created using the new operator:

```
$dog1 = new Dog(“BooBoo”, “Iwao”, “EnglishBulldog”);
print “Dog 1: {$dog1->getName()}\n;
```

This produces:
// Dog 1: BooBoo Iwao
//You see here that I can create a new project and even have properties there that I can have, but not use! Any arguments supplied are passed to the constructor. So in my example I pass the the first name, the last name, and the breed to the constructor. The constructor method uses the pseudo-variable $this to assign values to each of the object’s properties. A Dog object is now easier to instantiate and safer to use. Instantiation and setup are completed in a single statement. Any code that uses a Dog object can be reasonably sure that all its properties are initialized. This predictability is an important aspect of object-oriented programming. You should design your classes so that users of objects can be sure of their features. By the same token, when you use an object, you should be sure of its type.