

The code is lengthy. However, as pointed out previously, like the set methods, the get methods are very similar to each other. Once you have created one successful get method, you can copy and paste it to create the other methods. All that needs to change for each get method is the name of the method and the property being returned.

Example 3-11. The lab.php program using set and get methods

```
<?php
require_once("dog.php");
$lab = new Dog;
// -----Set Properties-----
$dog_error_message = $lab->set_dog_name('Fred');
print $dog_error_message == TRUE ? 'Name update successful<br/>' : 'Name update not
successful<br/>';
$dog_error_message = $lab->set_dog_weight(50);
print $dog_error_message == TRUE ? 'Weight update successful<br />' : 'Weight update not
successful<br />';

$dog_error_message = $lab->set_dog_breed('Lab');
print $dog_error_message == TRUE ? 'Breed update successful<br />' : 'Breed update not
successful<br />';
$dog_error_message = $lab->set_dog_color('Yellow');
print $dog_error_message == TRUE ? 'Color update successful<br />' : 'Color update not
successful<br />';
// -----Get Properties-----
print $lab->get_dog_name() . "<br/>";
print $lab->get_dog_weight() . "<br />";
print $lab->get_dog_breed() . "<br />";
print $lab->get_dog_color() . "<br />";
$dog_properties = $lab->get_properties();
list($dog_weight, $dog_breed, $dog_color) = explode(',', $dog_properties);
print "Dog weight is $dog_weight. Dog breed is $dog_breed. Dog color is $dog_color.";
?>
```

When viewing the code in Example 3-11, note that the print statements call the get methods (print `$lab->get_dog_name() . "
"`). The order of operations, which will be discussed in more detail in a later chapter, causes the method (`get_dog_name()`) to be executed first, even though normally a line of code would execute from left to right. The method returns the value in `dog_name` (the string "Fred"). The string is placed in the same location in which the method call was located. After the get method executes, the code line is now

```
print "Fred" . "<br/>";
```

Then the code line executes from left to right, producing the output

```
Fred <br/>
```

Programming note—Unlike properties, methods must NOT be included in quotes (" or "). The string must be broken apart and concatenated using the '.', as shown in the example. If a method is included IN quotes, PHP will display an error message.