Set Methods

This example is still very limited because you can't currently adjust the values in the properties to relate to the actual objects you created (such as `$lab`). In order to adjust these properties, you must have some ability to access the properties from the program that uses the object (`lab.php`). However, because of encapsulation and security concerns, you do not want to expose the properties to be directly manipulated by the calling program. Object-oriented programming standards require that you create your properties as "private" (as you have done already) and then use actual methods in the class to change any values.

> *Security and reliability—Creating set methods in classes provides the ability for the class to verify that the information that is to be placed in a property is valid before the property is updated. If this verification is not done before changing the value in a property corruption of data could take place. After the fact, it may be impossible or very difficult to correct invalid data that has been accepted.* `Set` *methods can reject invalid data and return error messages to the calling program.*

A `set` method allows values to be passed into the method. These values can then be verified before updating the properties in the object. Parameters (values) are passed into a method between the parentheses `()` in the method call.

```
$dog_error_message = $lab->set_dog_name('Fred');
```

If the `set_dog_name` method exists within the `Dog` class and accepts a string representing the name of the dog, you could use a method call similar to the previous code. This call would pass the string `"Fred"` into the `set_dog_name` method. It also provides the ability for the `set` method to return a value into the property `$dog_error_message` to indicate if the property was updated properly. You can simply pass a '`TRUE`' or 'FALSE' Boolean value back from the method to indicate the status of the update. The calling program then can determine how to handle the status of the update.

If you simply pass back a '`TRUE`' or '`FALSE`' you can use a simplified version of the PHP conditional statement, called the *ternary operator* to check `$dog_error_message`.

```
print $dog_error_message == TRUE ? 'Name update successful<br/>' : 'Name update not
successful<br/>';
```

> *For more information on the ternary conditional operator, visit*
>
> *http://php.net/manual/en/language.operators.comparison.php*
>
> *Security and performance—Use caution when display error messages to actual live users of your applications. You can provide too much information and expose your program code unnecessarily. Displaying a generic error message to the user may be a safer option. In live applications, log files should be created to record errors and access to the application itself.*

With this format, the calling program (`lab.php`) can easily determine the status of the update and display a corresponding message. The message between the `?` and the `:` ("`Name update successful`") will display if the string in `$dog_error_message` is '`TRUE`'. If the value in `$dog_error_message` is '`FALSE`' the string between the `:` and the `;` ("`Name update not successful`") will display.