

```

        $result = $lab->get_select($dog_app);
        print $result;
    }
} // try
catch(setException $e)
{
    echo $e->errorMessage();
}
catch(Exception $e)
{
    echo $e->getMessage();
}
catch(Error $e) // PHP 7+ only
{
    echo $e->getMessage();
}
?>

```

When comparing Example 4-12 to Example 5-6, the amount of code needed to handle exceptions is less than using `if/else` conditional statements. The logical flow of the program is easier to follow with very few `else` statements. This occurs because the exceptions thrown from all the files in this application are handled by the `catch` blocks in `dog_interface`. The user errors are thrown to a special `setException` exception. The system errors are captured by the `Error` catch block. The `error_check_dog_app` method (in Example 4-12) has been replaced by the `setException` class. The code in the class is very similar to the code in the `error_check_dog_app`. The display of individual update messages in the `$eMessage` string is removed, since this class reacts to user errors, not successful updates. A general `print` line has been added in the main body of the code to let the users know that all updates have been successful. The `try` block has been added around all the code in this interface. This helps to capture any problems in any part of this application. Notice that an exception is also thrown if the `dog_container` file cannot be found.

Only three `catch` blocks are required for this application. The `setException` `catch` block calls the `errorMessage` method from the `setException` class, which determines what user errors have occurred. The information is then displayed back to the user. The `Exception` `catch` block handles all other exceptions. It currently displays this information to the user. However, the `Exception` and `Error` `catch` blocks are currently providing the user too much information. It is a violation of security to inform the user what other problems the application maybe experiencing. You should just tell them that the system is not currently available and ask them to check back later. Displaying detailed errors is okay when you're testing. However, it's not good for the real world. You will resolve this breach of security in the next section.

*For more information on exception handling, visit*

*Examples: [http://www.w3schools.com/php/php\\_exception.asp](http://www.w3schools.com/php/php_exception.asp)*

*Video: <https://www.thenewboston.com/videos.php?cat=11&video=17171>*