The `testerror` class (in Example 5-1) includes a method to cause an error (`produceerror`) and a method that throws an exception (`throwexception`). However, the class does not have `try` or `catch` blocks. It does not have the ability to react to any exceptions or errors that might occur.

The `handleerror` program (in Example 5-2) includes a method that will handle user errors (`errorHandler`), along with the `set_error_handler` command to redirect errors to this method. It also includes a class (`userException`) that can react when the `userException` exception is thrown in the `try` block. The `require_once` statement is included in the `try` block in an attempt to capture the error if the file is missing. However, this happens to be a system error (not a user error) which cannot be redirected. To capture system errors in PHP 7, the Error class must be used within a `catch` block as previously shown.

After the `require_once` statement, an instance of class `testerror` is created. If this class is missing, the system will also error with a fatal message. The block calls the `produceerror` method, which causes a user error. This error is redirected to the `errorHandler`, which throws an exception (`errorException`). The `catch` block receives the exception and displays the error message. Since exceptions do not shut down the program (like fatal errors), the flow of the program jumps to the first line after all the `catch` blocks and executes the echo statement (`echo "This line will display";`). The reaction to the error will cause the program to skip any remaining code in the `try` block. In this example, the `throwexception` method call would be ignored.

If the `$tester->produceerror()` line is commented out, the `throwexception` method call can take place. The `userException` is thrown in the method. The `userException` class inherits the `Exception` class. No special methods have been included in `userException`. The flow of the program will jump to the `catch` block for `userException`. This block uses the `Exception` class `getMessage` method to display the message. The logic then jumps to the first line of code after the `catch` blocks and executes the `echo "This line will display"` statement.

Program note—`try/catch` can also include a `finally` block after all `catch` blocks. The `finally` block will execute for all caught exceptions after the associated `catch` block has executed. PHP allows the `finally` block to exist without any `catch` blocks (but the `try` block must still exist). One of the most common uses of the `finally` block is to close files and/or databases when an exception has occurred. A program should not close before files and databases have been properly closed. If not closed properly, the data may become corrupt and not be accessible.

Do It

1. Go to the book's web site and download the files for Examples 5-1 and 5-2. Adjust the `testerror` program to only create an error. Create an additional `testexception` program (with a `testexception` class) to throw an exception. Now adjust the `handleerror` program to create an instance of both programs. The `handleerror` program should now be able to handle errors or exceptions from either program (class).

Exception and Error Handling vs. If/Else Conditions

A programmer can always choose to handle exceptions and errors using `If/else` conditional statements as shown in the dog application files from Chapter 4. It is not any less efficient to handle errors in this way (it might even be more efficient). However, as you are about to discover, the *attitude* of the code in the business rules tier (and data tier) changes if you use exception handling. When you use `if/else` statements, the flow of the program spends a lot of time being pessimistic by preparing for the worst (errors and/or exceptions). In many cases, by using exception handling, the coding for most of the business rules tier (and data tier) becomes optimistic including code which handles the normal operation of the program. The application relies on the interface tier to handle any problems.

www.it-ebooks.info