

7. The `!` operator reverses a `TRUE` result to a `FALSE` result.
8. `print_f` can be used to display the contents of an array.
9. `Try/Catch` blocks should reside in the business rules and data tiers but not the interface tier.
10. Only `Exceptions` intentionally thrown by program code should be caught.

Short Answer/Essay

1. Explain how hierarchy of exception handling works with three-tier applications.
2. What is the difference between an error and an exception?
3. How do you correct the time zone if PHP is providing an incorrect timestamp?
4. How can PHP programmers try to capture errors so they can be treated as if they are exceptions?
5. Why is it important to have multiple log files produced by an application?

Projects

1. Adjust the code from project #1 (or #2) from Chapter 4 to include exception handling and logging.
2. Create an application that will register contestants for your favorite game show. Include verification of the data entered using HTML5 and JavaScript. Also validate the data when it is passed to the application. The application should include an interface php program (interface tier) and a registration class (business rules tier). The registration class should include exception handling (both user exceptions and program exceptions). The interface program should handle the exceptions using `try/catch` blocks as shown in this chapter.

Term Project

1. Update the ABC Computer Parts Warehouse Inventory application to include exception handling. The application should attempt to handle all exceptions, and errors, when possible.

User exceptions should be logged to a user log. All other exceptions should be logged to a system log. If the exception is considered to be extreme (will cause the program to otherwise crash), an e-mail should be sent to the system administrator. Hint: The `Try/Catch` block should only exist in the interface tier.