Chapter 7

The Exception Class

Exceptions

- Unified method of error handling
  - Divide code into small *try* blocks
    - Code is executed here and error might occur
  - *Throw* each possible error
    - Runtime errors
  - Write *catch* cases for each possible error to handle the error

Exceptions: Example

```
try{
  $fp = fopen(‘spork.txt’, ‘r’);
  if (!$fp){
              throw new Exception(‘What you say!’,
  4);
  }
  echo ‘everything’s fine!<br />’;
}
catch (Exception $e){
  echo “Exception “.$e–>getCode().” “.$e–
  >getMessage().”;
}
```

Exception Class: Methods

- Exception is a built-in object with various methods
  - getCode() returns the exception code
  - getMessage() returns the exception message
  - getFile() returns the path to the code file
  - getLine() returns the line number of the exception
  - getTrace() returns a backtrace
  - getTraceAsString() returns a backtrace formatted as a string
  - __toString() prints the entire thing

User-Defined Exceptions

- Can pass objects you like
- You extend the Exception class to create your own exception class
- P. 196 Listing 7.2 Exception class
- P. 197 Listing 7.3 user_definfed_exception.php
  - User defined exception class

Bob’s Auto Parts

- Things that can go wrong
  - File cannot be opened
  - A lock cannot be obtained
  - File cannot be written to
- Exception class to handle this
  - P. 199 listing 7.4 file_exceptions.php
  - P. 200-202 Listing 7.5 processorder.php rewritten to use exception class