

This example might produce an exception if `$secondNumber` contains a zero (dividing by zero). If the exception occurs, the code would jump to the catch block. Any code in the block will then be executed. The statement `$e->getMessage();` will display any system message related to the exception (in this case a message about the attempt to divide by zero). However, you do not have to use the system message; you can use `echo` or `print` to display messages to the users.

```
try {
$result = $firstNumber /$secondNumber;
}
catch(Exception $e) {
echo "You entered zero for the second number. Your entry must be greater than zero";
}
```

However, there is a problem with these examples. If you were trying to catch more than one type of exception in the try block, all exceptions would go into the one catch block. Any exception would display the same message. There are a couple of different ways you can handle this.

One way is by throwing your own exception instead of having the system throw it.

```
try {
    if ($secondNumber == 0)
    { throw new Exception("Zero Exception"); }
    else { $result = $firstnumber / $secondnumber; }
    // other code with exceptions }
catch(Exception $e) {
    switch ($e->getMessage()) {
        case "Zero Exception":
            echo "The value of second number must be greater than zero";
            break;
        case "Some other exception":
            echo "You did something else wrong";
            break;
        default:
            echo $e->getMessage();
    }
}
```

*Programming note—In addition to `getMessage` method, the `Exception` and `Error` objects include:*

*`getCode()`—Displays the code causing the exception*

*`getFile()`—Displays the file name containing code that threw the exception*

*`getLine()`—Displays the line number that threw the exception*

*`getTrace()` and `getTraceAsString()`—Displays backtrace (exception flow through the program) information*

*In some circumstances it might be appropriate to display the `Exception` or `Error` message to the users. However, the other methods should only be used for debugging or log entries. Providing code information to the users is usually unnecessary and is a breach of security.*