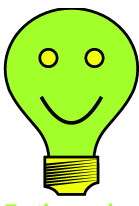
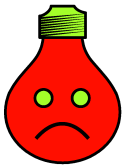


Debugging
sucks.



Testing rocks.

Testing on the Toilet

Extracting Helper Classes

Month day, 2007

In a previous episode, we **extracted methods to simplify testing** in Python. But if these extracted methods make the most sense as private class members, how can you write your production code so it doesn't depend on your test code? In Python this is easy; but in C++, testing private members requires more **friend** contortions than a game of Twister™.

```
// my_package/dashboard.h
class Dashboard {
private:
    scoped_ptr<Database> database_; // instantiated in constructor
    // Declaration of functions GetResults(), GetResultsFromCache(),
    // GetResultsFromDatabase(), CountPassFail()
    friend class DashboardTest; // one friend declaration per test fixture
};
```

You can apply the **Extract Class** and **Extract Interface** refactorings to create a new **helper class** containing the implementation. Forward declare the new interface in the .h of the original class, and have the original class hold a pointer to the interface. (This is similar to the Pimpl idiom.) You can distinguish between the public API and the implementation details by separating the headers into different subdirectories (/my_package/public/ and /my_package/ in this example):

```
// my_package/public/dashboard.h
class ResultsLog; // extracted helper interface
class Dashboard {
public:
    explicit Dashboard(ResultsLog* results) : results_(results) { }
private:
    scoped_ptr<ResultsLog> results_;
};

// my_package/results_log.h
class ResultsLog {
public:
    // Declaration of functions GetResults(), GetResultsFromCache(),
    // GetResultsFromDatabase(), CountPassFail()
};

// my_package/live_results_log.h
class LiveResultsLog : public ResultsLog {
public:
    explicit LiveResultsLog(Database* database) : database_(database) { }
};
```

Now you can test **LiveResultsLog** without resorting to **friend** declarations. This also enables you to inject **MockResultsLog** instance when testing the Dashboard class. The functionality is still private to the original class, and the use of a helper class results in smaller classes with better-defined responsibilities.

More information, discussion, and archives:
<http://googletesting.blogspot.com>



Copyright © 2007 Google, Inc. Licensed under a Creative Commons
Attribution-ShareAlike 2.5 License (<http://creativecommons.org/licenses/by-sa/2.5/>).

