The University of Sydney
School of Information Technologies

Software Quality Assurance: SOFT3302

Tutorial — Week 6

Objectives

This tutorial gives experience designing unit tests using a unit-testing framework. (You are free to do this tutorial in Python, C++ or Java.) By the end of this tutorial you should be able to write a suite of test cases and have an understanding of this processes as well as experience in running the test and displaying the results using a test runner.

Prework

Review the use of unit testing frameworks from the lecture.

Consider the short specification of a class below. You will be writing this class and unit tests for this class in the lab this week.

Write a 'queue of integers' class. The constructor should take one argument: the maximum size of the queue. The default constructor should create a queue of 10 elements. The queue should have the usual push/pop semantics. An attempt to pop from an empty queue should raise an exception. An attempt to push onto a full queue should also raise an exception. The queue should have a size method that returns the number of elements in the queue and an empty method that returns true/false depending on whether or not the queue is empty/non-empty.

Labwork

Write an implementation of the above queue class using test-drive development. *Remember, this means write the tests first and then code to pass the tests.* Run the tests you develop using a text test runner and a graphical test runner if available.

If the queue class is implemented as a wrapper/extension of another class, such as a deque, which presumably is 'tested' and is low-risk/high confidence, is there a need to test the new class?