D – Testing and Debugging

# Overview

This topic introduces the concepts of testing and verifying that code is running correctly. Two approaches will be presented and compared: Ad-hoc test drivers and Unit Tests.

Test drivers that rely on user input/output are the traditional means of verifying that code runs properly. The biggest benefit of test drivers is that they provide quick "ad-hoc" tests and are simple to produce on any IDE. However, test drivers have some major downfalls.

Current programming techniques make use of Unit Tests. A unit test is a simple, isolated test of some part of a class. Unit tests provide a way to clearly show the success or failure of a particular section of code. Unit tests are a key part of Test Driven Development (TDD).

Regardless of whether the developer uses test drivers or unit tests, it is important to remember that the developer is still responsible to make sure that the tests (and thus, the test results) are valid. Improper use of test drivers and unit tests can lead to

* "false positives" - indicating that something works when it really doesn't, and
* "false negatives" - indicating that something is broken when it really isn't

***Note****: For this course, students will be provided with unit tests and will not be expected to have to write their own unit tests. The unit tests supplied to students will use* ***nUnit 2.5.7*** *which is already installed in the lab. To use these libraries at home, simply install* [***nUnit 2.5.7***](http://launchpad.net/nunitv2/2.5/2.5.7/+download/NUnit-2.5.7.10213.msi) *on your home computer.*

**Daily LOGs**

The following daily LOGs are covered in this package.

**General Programming Concepts and Terms**

* Explain the role of testing as it applies to software development
* Define and compare the terms "compile-time error" and "run-time error"
* Define the term "test driver"
* Create simple ad-hoc test drivers to test for run-time errors
* Define the term "TDD"
* Compare and contrast Test Drivers and Unit Tests
* Define the terms "false positive" and "false negative"
* List three downfalls of using Test Drivers
* Identify four benefits of using Unit Tests
* Add unit testing libraries and unit testing packages to existing programs
* Use unit tests to validate the requirements of a class’ design and behaviour
* Diagnose and correct software problems by using unit tests