References

Brian Marick’s Testing Web Site: http://www.testing.com/

Testing for Programmers, Brian Marick, Available at: http://www.testing.com/writings.html
Testing

Johnson's Law

If it is not tested it does not work

Beck’s Law

It is is not tested it does not exist
The more time between coding and testing

- More effort is needed to write tests
- More effort is needed to find bugs
- Fewer bugs are found
- Time is wasted working with buggy code
- Development time increases
- Quality decreases
Unit Testing

Tests individual code segments

Automotated tests
What wrong with:

- Using print statements
- Writing driver program in main
- Writing small sample programs to run code
- Running program and testing it be using it
We have a QA Team, so why should I write tests?
When to Write Tests

First write the tests

Then write the code to be tested

Writing tests first saves time

Makes you clear of the interface & functionality of the code

Removes temptation to skip tests
Can you test every line of code?
If you can should you test every line of code?
What to Test

Everything that could possibly break

Test values
- Inside valid range
- Outside valid range
- On the boundary between valid/invalid

GUIs are very hard to test
- Keep GUI layer very thin
- Unit test program behind the GUI, not the GUI
Common Things Programs Handle Incorrectly

Adapted with permission from “A Short Catalog of Test Ideas” by Brian Marick,
http://www.testing.com/writings.html

Strings
Empty String

Collections
Empty Collection
Collection with one element
Collection with duplicate elements
Collections with maximum possible size

Numbers
Zero
The smallest number
Just below the smallest number
The largest number
Just above the largest number
Personal Common Error Catalog

Keep a log of your common errors