Producing Production Quality Software

Lecture 5b: Testing Limitations
Prof. Arthur P. Goldberg
Fall, 2005
Testing Limitations

• Naturally, cannot
  – Prove code correct
  – Find all bugs
  – Examine all input combinations

• Might not
  – Execute all lines of code

• Unit and integration test typically do not examine
  – Performance
  – Memory use
  – Synchronization, timing
Performance Testing

• A whole course
• Basics
  – Vary conditions and input
    • Equipment (computing resources)
    • Input
      – Size(s)
      – Characteristics
  – Monitor performance
  – Model too, perhaps
– See
  • Daniel A. Menasce (Author), Virgilio A.F. Almeida, Capacity Planning for Web Services: Metrics, Models, and Methods
Simple Performance Test Example

• Take an O(n log(n)) algorithm
• Run it with a range of input sizes
• Be careful
  – Use appropriately distributed (probably random) inputs
  – Don’t let it page
  – Avoid initialization issues—measure run time for each input size multiple times
  – Watch out for contention
• Observe whether the run times grow more slowly than n log(n)
Memory Leaks

• When long-running programs consume growing amounts of VM
  – E. g., Mozilla
  – Memory leak detection
    • Compare
      – Memory in use
      – Memory that should be in use
      – Some tools will show java memory use
    • Trace
      – New – null it
• Leak avoidance in ArrayList.java
• Digression: how does gc work?
C memory leaks

– malloc – free
– See Gray Watson’s “Debug Malloc Library”
– Replaces malloc, and free
– Provide debugging facilities
  • Memory-leak tracking
  • Fence-post write detection
  • File/line number reporting
  • General logging of statistics
Synchronization, timing

• AKA ‘race conditions’ or ‘Heisenbug’
  – (Werner Heisenberg and the Uncertainty Principle)
• Difficult or almost impossible to detect with testing
  – Depends on relative timing of concurrent execution
  – Extremely difficult to control and reproduce
• Therefore, write the synchronization code correctly!
• We’ll spend an evening on concurrent code