Software Testing

Course Overview

Shih-Kun Huang skhuang@nycu.edu.tw

Outline

- Software Testing
  - Paul Ammann & Jeff Offutt
  - A craftsman's approach
  - Foundations
- Testing & Secure Programming
  - The Fuzzing Book
- Common Terms
- Testing @ Microsoft
- Testing @ Google
- Testing @ NYCU (Testing Your Own Projects)

Common Terms of Software Testing

- Test Coverage
  - Ammann-Offutt's Part 2
  - Jorgensen's, path testing
- Test-Driven Development (TDD) on testing
  - framework and JUnit related
- Combinatorial testing (all pairs-testing)
- Fuzzing (evaluating test cases)
- Continuous Integration

Overview

- Foundation
- Coverage Criteria
- Testing in Practice
- Security Testing

Foundations

- Model-Driven Test Design
- Test Automation
- Putting Testing First
- Crieria-Based Test Design

Coverage Criteria

- Input Space Partitioning
- Graph Coverage
- Logic Coverage
- Sytax-Based Testing

Testing in Practice

- Managing the Test Process
- Writing the Test process
- Writing Test Plans
- Test Implementation
- Regression Testing for Evolving Software
- Writing Effective Test Oracles

Security Testing

- Fuzz Testing
- Symbolic Testing

Grading Policy

- Lab and Homework (60%)
  - Unit test - Java
  - Coverage tools
  - Stub / Mock
  - Selenium (web)
  - JMeter (web)
  - CI & Online coverage (git)
  - AFL (auto)
  - Symbolic execution (auto)
- Challenge Work (10%)
  - Report 10 bug issues, graded according to the project's popularity and bug severity
- Term Project (30%)
  - fuzzing, unit testing, or any testing related techniques
  - project presentation
  - report

Candidate Projects

- a github project to add testing to an open source project
- fuzz testing (afl)
  - bug bounty, CVE
- symbolic testing
  - klee, triton, angr, s2e
- triage testing (!exploitable)
- combinatorial testing
- metamorphic testing
- Projects of Your Jobs (戊組、丁組)