

Exceptions, Errors, and all that gunk

Today

- Administrivia
- RI reviews back
- Testing Tips II
- Design exercise, revisited
- Exceptions, Errors, Throwables, and all that

Administrivia

- No office hours for Prof. Lane this week
- Jason is here and available
- Midterm: Wed, Mar 9
- Jason will proctor (lucky guy, isn't he?)
- R1 writeups back today
- R2 assigned today
- Due Mar 23 (Wed after spring break)
- Kohno, T., et al., "Analysis of an Electronic Voting System." IEEE Symposium on Security and Privacy, Oakland, CA, May, 2004.

Testing Tips II

- But which test cases do I need?
- ≥ 2 schools of thought:
- **Code coverage**: a test case that forces the code to execute every possible alternative of every branch
 - Eek!
 - In theory: every time you write a condition (if, while, switch, etc.), write a test case for either branch
 - In practice: write tests for significant branches -- esp ones handling input
 - Sort-of doable for unit test. Doesn't scale. Also doesn't catch all cases

Testing Tips II

- **Case coverage:** start by writing down set of *use cases* that cover all expected applications of software
- Write a test (or more) for each use case
- Differences in cases will hit significant code paths
- If you find a bug, create a special use case/test that exercises that bug