2

- Fault, Error, Failure
- State of a program (wrong vs. expected)
- RIP Fault Model (fault to failure)
- Dealing with Faults
- Testing vs. Debugging

3

- QUESTION: given faulty program, identify stuff
- Ways of validating a test suite, line intersect example

4

- Static Testing
- Dynamic Testing (black-box, white-box)
- Naughty words
- When to stop testing?
- Test Case
    - Observability
    - Controllability
    - Anatomy
    - Test Set, Script
- Coverage, Coverage Level
- Test Requirements

5

- Exploratory Testing
    - When it does well
    - The process
    - Output from process

6

- CFG intro
- Steps in compilation
- Abstract Syntax Tree (AST)

7

- CFG
    - Basic block
    - If statements
    - While statements
    - For statements

8

- Large CFG examples
- Test Path Definition
- Coverage definition
- Test Case -> Test Path mapping
- Test case, test path examples
- Deterministic vs. Nondeterministic methods
- Statement coverage definition and example
- Branch coverage definition and example

9

- Complete Path Coverage criterion
- FMS definition
    - Node coverage
    - Edge coverage
    - Edge-pair coverage
    - Round trip path
    - Simple Round Trip Coverage (SRTC)
    - Complete Round Trip Coverage (CRTC)
    - Deriving FSM (best with a modelling state)
    - Advantages of FSMs
    - Disadvantages

10

- Applications of CFGs (input-space, program-based)
- Generating Inputs
    - Using Grammars vs. RegExps
    - Recognizers vs. Generators
    - Grammar Mutation Operators
        - Nonterminal Replacement
        - Terminal Replacement
        - Terminal and Nonterminal Deletion
        - Terminal and Nonterminal Duplication
    - Using Grammar Mutation Operators, and why?

11

- Fuzzing definition + back story
    - Generation-based
        - HTML5 example
        - C example
    - Mutation-based
    - Advantages + Disadvantages
- Chaos Monkey story

13 (no 12)

- How much testing is enough? Conclusions at end
- JUnit case study
    - Causes of uncovered code (too simple, empty method, etc.)

14

- Mutation Testing
    - Ground String
    - Mutation Operator
    - Mutant
    - Killing mutants
    - Mutation score
    - Example Mutant, and at end
    - Uninteresting Mutants (stillborn, trivial, etc.)
- GOALS of Mutation Testing

15

- Strong Mutants
- Weak Mutants
- Strong Mutation Coverage (SMC)
- Weak Mutation Coverage (WMC)
- Examples of Strong and Weak Mutations
- Mutation testing workflow diagram
- Integration Mutation
- A bunch of example mutation operators
- Mutation for OO Programs
- Exercise: Killing mutants with test cases
- Syntax-Based Testing summary chart

16

- Is mutation testing any good? Case Study
- Is coverage at good means for testing?

17

- Engineering Test Suites
- Not delaying the writing of test cases

18

- Reason to test: avoiding regressions
- Test Design Principles
    - Many small tests
    - Easy to add tests
    - Unit vs. Integration