CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Exercise 5: Mutation Testing

Gregory Gay
DIT635 - February 28, 2020

Finish In-Class Activity First!

The Planning System Returns

- Code on Canvas:
  - https://canvas.gu.se/courses/25762/files/folder/Misc?preview=2280199
- Everybody likes meetings.
  - Not true - but we need to book them.
- We don’t want to double-book rooms or employees for meetings.
- System to manage schedules and meetings.

Mutate the Meeting Planner

- Create at least four mutants for classes from the MeetingPlanner system.
  - Create at least one from each category:
    - invalid (doesn’t compile)
    - valid-but-not-useful (compiles, fails for any test case)
    - useful
    - equivalent (no test will ever fail)
  - Use different operators for each mutant
    - 1+ from each category in handout.
  - Try mutating different parts of the code.

Assess Your Test Cases

- Run the functional and structural tests you created previously. Do they detect the mutants?
  - If not, create new test cases that will detect them.
  - If equivalent, make sure you understand why the mutant will never be detected.
- Identify the minimal set of your tests that will detect all four mutants.
- If you finish quickly, try this for the CoffeeMaker.
  - (part of Assignment 3)

UNIVERSITY OF GOTHENBURG

CHALMERS

UNIVERSITY OF TECHNOLOGY