Mutations Testing

- Link to get started is on the course Calendar: https://courses.cs.washington.edu/courses/cse403/25wi/project/mutation-assignment.html
- We encourage you to use **attu** to complete this assignment.
- You may work with a partner – **Please join one of the existing Canvas groups**
- There is a **README** and an **ASSIGNMENT** file in the repository. Read both of these!
- ASSIGNMENT.md includes some background information about mutation testing which is helpful; see also slides from Feb 10 lecture

Review

Mutation testing is a way of evaluating a test suite

- It tells you how good your suite is, and it helps you improve your test suite.

A mutant is a variation of a program

- Equivalent mutant returns the same value as the program
- Non-equivalent mutant returns a different value for some input

A test suite that runs against a program also can be run against a mutant

A mutant is detected/killed if the mutant fails the test suite

- Otherwise the mutant is live

Goal is for the test suite to detect/kill all the non-equivalent mutants

- If one is live, it means a bug can exist in the program without the tests catching it