CSC 217 Lab 02

Software Engineering Best Practices

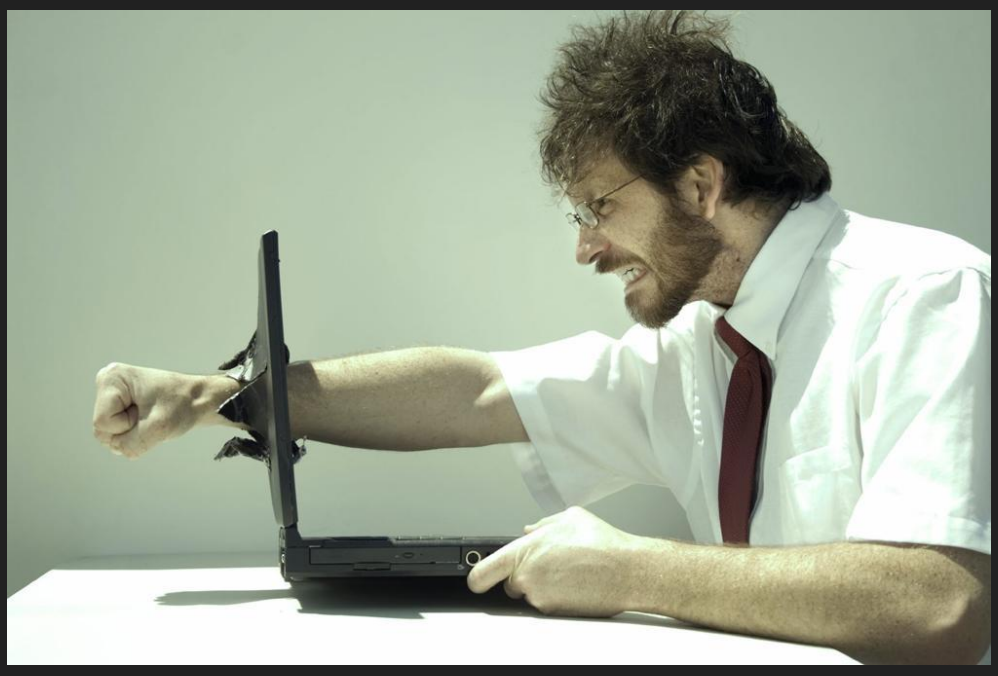
Lab Overview

- Deadlines & Reminders
- Debugging
- Software Engineering Best Practices in PackScheduler
  - Set up Static Analysis tools to run automatically
  - Write JUnit tests to achieve > 80% statement/line coverage for:
    - Student
    - StudentRecordIO
    - StudentDirectory
  - Write and run black box tests on StudentDirectoryGUI
- Lab Wrap-Up

Deadlines and Reminders

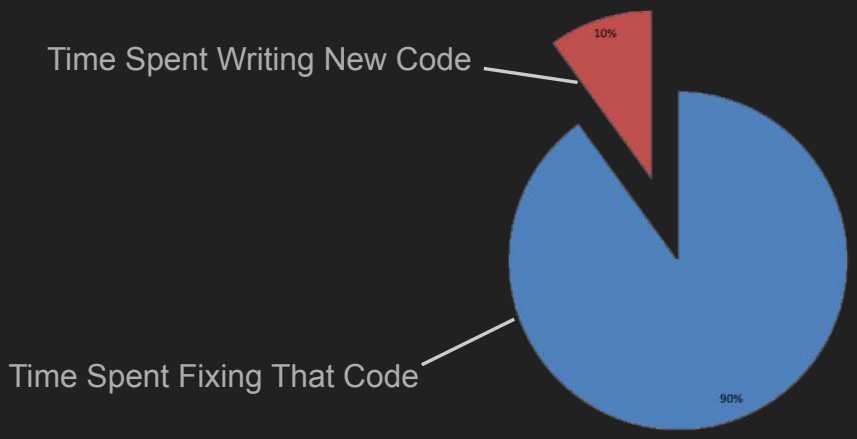
- Help-Seeking Reminders
  - Attend office hours (must create ticket on MDH)
  - Piazza post quality
    - Include repository
    - State the failing test / method
    - Describe how you've tried to debug the problem or how you've tried to localize the problem to a few lines of code.

Debugging!



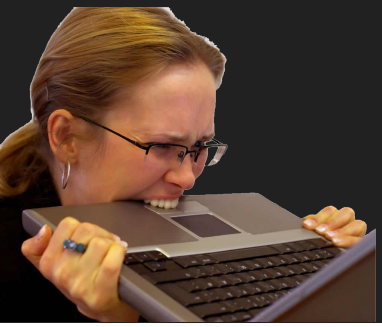
What is Debugging?

- Stepping through code one line at a time to troubleshoot problems
- The **MAJORITY** of software development is testing, debugging and bug-fixing



Why Do Bugs Happen?

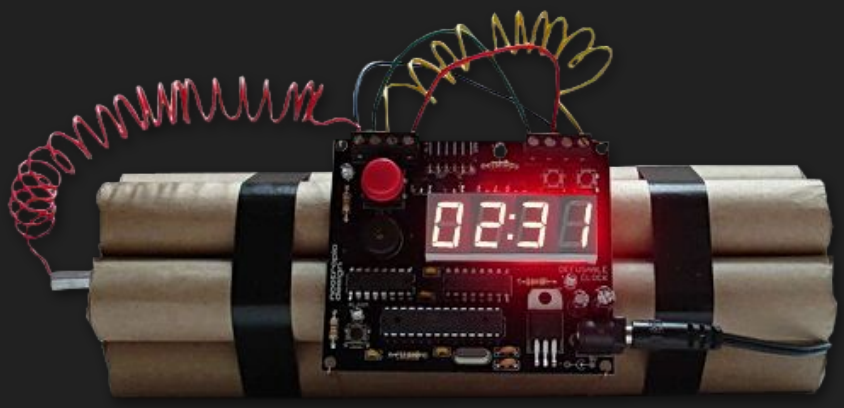
- OS problem? Compiler? Hardware? (very very unlikely)
- Unclear requirements / specifications, constantly changing, unreasonable schedules
- Lack of mastery of programming tools / language
- Inadequate testing procedures
- Faulty logic



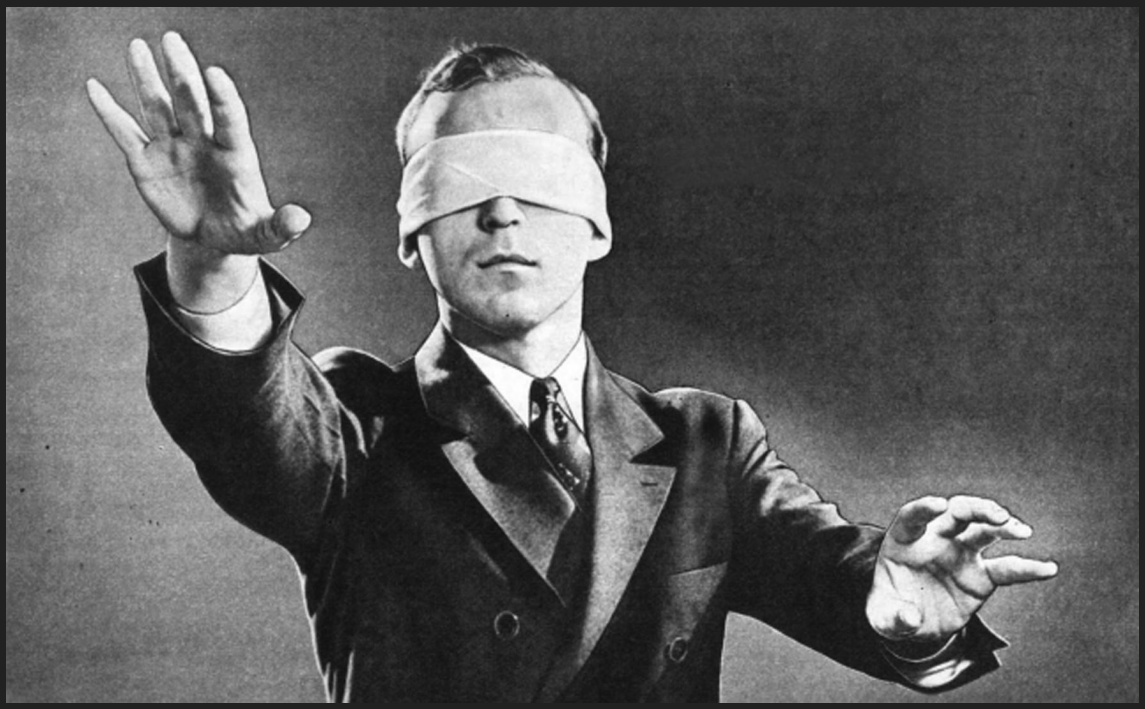
Addressed in this course

Debugging Approaches

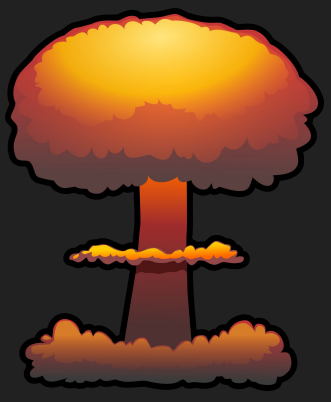
- Bugs are like ticking time-bombs in your code.
- Need to be defused/fixed before they “blow-up” in a user’s face



Just change things and hope it works!

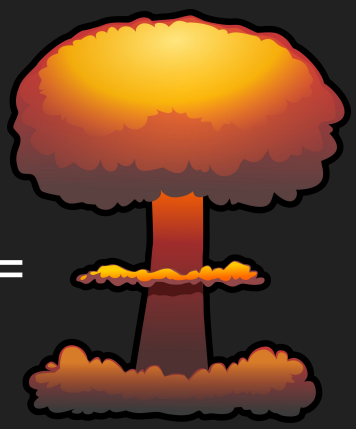


=



Add lots of print statements, and test ideas!

=



or

Use a Debugger

=

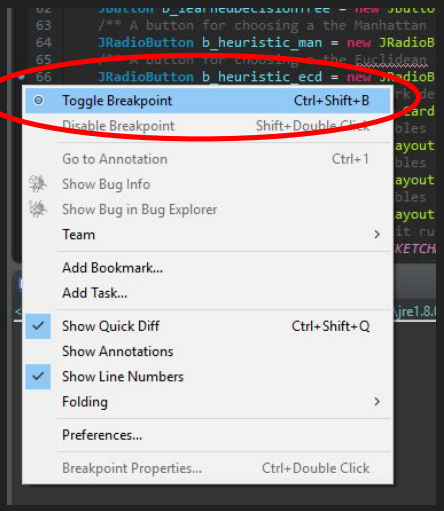
Source Level Debugging

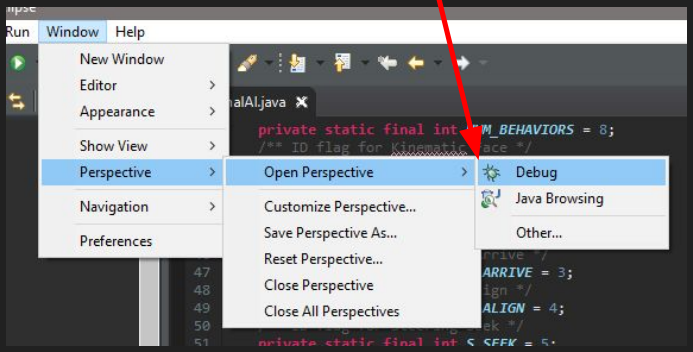
- Symbolic debugging lets you single step through program, and modify/examine variables while program executes.
- Lets you suspend program when errors/exceptions happen and allows you to trace back where the problem came from.
- Built into most IDEs.

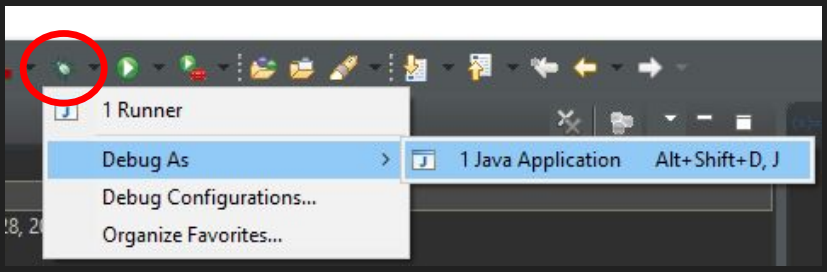
Slides - Dr Tyler Bletsch : tkbletsc@ncsu.edu

Debugging in Eclipse

- Anything that can be run in eclipse can be debugged
- Add breakpoints to stop your code at certain points.
- The debugging perspective.

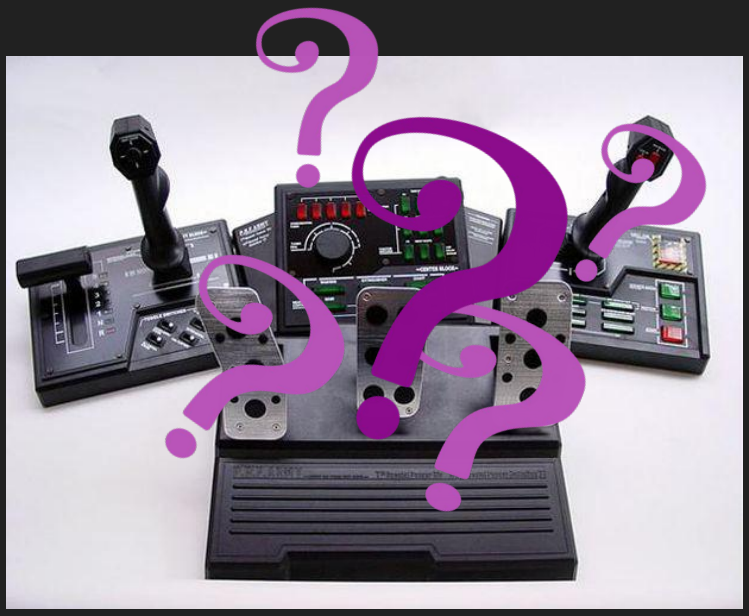






Eclipse Debugging Controls:

- Continue 
- Suspend 
- Stop 
- Step into 
- Step over 

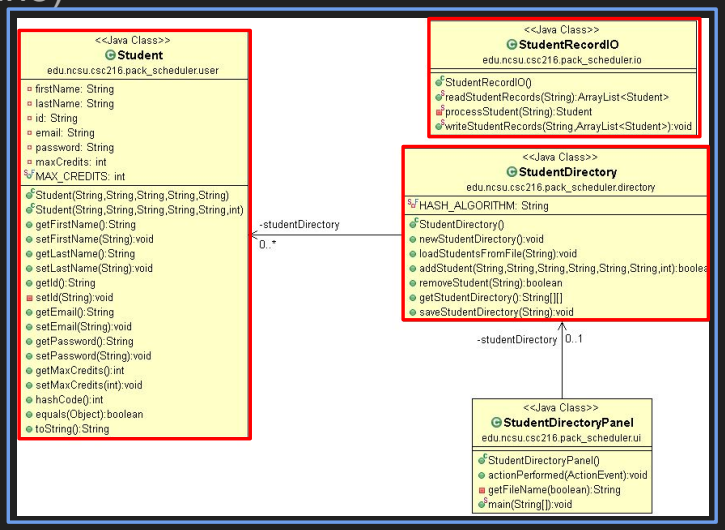


PackScheduler - Testing

- PackScheduler is our lab project - we're writing a course registration system!
- Unit test Student, StudentRecordIO, and Student Directory (red outlines)
- System test the whole project (blue outline)

Today's Focus

- Testing!!!
- JUnit tests for > 80% line coverage
- System testing the whole program



Lab Overview

Lab activities are in Moodle

- Team: Pull the latest from the repo so that you're starting from a common set of code
- Write TEST code…
- … while pair programming!
- If you have any technology questions, now is the time to troubleshoot!

Extra Credit Opportunity!

Extra Credit

Grade Item	Points	Details
Over 90% Statement Coverage	1-3	One point of extra credit for each class with over 90% statement coverage.
100% Statement Coverage	1-3	One point of extra credit for each class with 100% statement coverage except for the paths described in the writeup (in addition to the points earned for the 90% statement threshold).
100% Condition Coverage	1-3	One point of extra credit for each class with 100% condition coverage except for paths described in the writeup (in addition to the points earned for the 90% & 100% statement threshold).

Wrap-Up

General Wrap-Up

- Deadline Reminder (see board)
- Exchange contact information with your partner
- Make a plan for finishing up the lab & record in README.md

Participation Outside of Lab (Guess which the teaching staff prefer?)

- If you pair program, note that in the commit comments so everyone gets credit!
- If you split the work, at least one contribution by each partner

REMINDER: We are expecting a significant contribution from all team members outside of lab!

Record Tasks & Owners

Tasks only get done when someone owns them!

Identify the tasks required to complete Lab 2

- Edit README.md to list the tasks required to complete Lab 2 (at top of README - should come before Lab 1 tasks)
- Add an owner to each task
- Add a deadline to each task

Deadlines should be at least 48 hours before the lab deadline so team members can help out and finish the lab if a team member runs into issues.

Notify team early if you run into problems with your tasks!