Lab 2: Intro Material
Software Engineering and Pair Programming

Notes for CSC 100 - The Beauty and Joy of Computing
The University of North Carolina at Greensboro

Question to think about….

In these labs you are writing programs

How does this relate to professional software development?

Software Engineering
How to control the complex process of creating software

Traditional "software lifecycle" has well defined phases that feed into each other one-way - called the "Waterfall Model"

Requirements definition and analysis
System design
Implementation
Testing
Operations and maintenance

In reality it never really works this way - early stages change even when working on later stages.

This is what we do in labs.
Software Engineering
How to control the complex process of creating software

Traditional "software lifecycle" has well defined phases that feed into each other one-way - called the "Waterfall Model"

Requirements definition and analysis
System design
Implementation
Testing
Operations and maintenance

In reality it never really works this way - early stages change even when working on later stages.

Recent trend: "Agile Development"
Among other things, testing done concurrently with development.
Idea: Faster error identification and correction - higher quality software?

Peer Reviews and Pair Programming

Two ideas for software development...

Peer Reviews: At regular intervals, a developer presents and explains their code to co-workers, who critically review code. Similar to a writer going over drafts with an editor.

Pair Programming: Development is actually a collaborative activity - pushes "review" so far back that it is simultaneous with development!

Pair Programming for Learning

From August 2013 Communications of the ACM:

Plug: Don't forget our UNCG ACM chapter!

Surprise!
We don't do pair programming because we like to torture you and stick you with terrible partners.
We do it because it helps people learn!
**Pair Programming: What to do**

Really simple concept:
- Two students, one computer
  - Roles: “Driver” and “Navigator”
    - Driver has keyboard/mouse, but navigator describes how to build solution
- Both students always active
  - Not “Driver” and “Sleepy passenger”
- Switch roles regularly
  - At least once per lab activity, if not more often
- Be open and respectful
  - If you don’t like a proposed solution, your job is to either explain why it’s not a good solution or to make the case for a better one - don’t just dismiss it!

**Details of Pair Programming in CSC 100**

At the beginning of lab:
- Check partner/workstation information sheet posted on lab door
- Find your workstation
- If you don’t know your partner, introduce yourself!
- Only one of you will log in to the workstation - decide which one (maybe first to sit down?) and log in to Snap! Cloud account
- If your partner is a no-show by 10:02, let the instructor or lab assistant know

Submitting your work:
- Submit activities as usual through “The G”
- One of you logs in to Canvas and enters names that worked together
- Everyone still does the quiz individually (before the next class)