Seeing vs. Doing in Computing Courses

Eric J. Schwabe
College of CDM
May 6th, 2011
Open with an Epigraph or Joke:

“I hear and I forget. I listen and I understand. I do and I remember.”

– Ancient Chinese wisdom

“I check my messages and – oh – uh – wait – What did he just say?”

– Modern local behavior
Teaching “How To ...”:

- In computing classes, I’m usually teaching students how to do something:
  - Design a database
  - Build a website
  - Write a program
  - Analyze an algorithm

- Homework feedback loop seemed too slow, so I began adding various kinds of practice
  - …modifying the balance between seeing and doing
Many Different Possibilities:

- Full lectures, then homework assignments
- Alternating lectures and lab sessions
- Lectures with occasional pauses for long group exercises
- Lectures with frequent pauses for short “checkpoint” exercises
- Lectures in a lab, with students at computers following along
- Extended group exercises with little preliminary lecturing
What I’ve Tried:

- Combinatorics / Discrete Mathematics:
  - Short pauses to work out examples
- Introductory Programming:
  - Short pauses to complete code fragments
  - Added 3–4 lab sessions per quarter
- Internet and Web:
  - 3–4 lab sessions per quarter
  - Expanded to alternating lectures and lab sessions
- Database Design:
  - Longer in-class exercises (started individually, then completed in groups)
What I’ve Found:

Challenges:
- Exercises must be sufficiently focused
- Real-life examples can get messy
- Groups may need guidance during longer exercises
- You may have to address other information sources
- Involving online students in group exercises is tricky

Benefits:
- Students can better assess their understanding
- Context and motivation are supplied frequently
- Students tend to be more attentive