Welcome to Com S 227

Please turn off your cell phone!
Welcome (or welcome back)

• Sorry!
  – You look familiar, but I can’t remember your name!
    • It could not possibly be because of old age

• Well, stop by my office sometime and introduce yourself
This Is Not A Powerpoint Slide

• I never use powerpoint slides
• I hate them
Who am I?

- Steve?

- Steve! [http://www.youtube.com/watch?v=ZOOLCh1K1og](http://www.youtube.com/watch?v=ZOOLCh1K1og)
But seriously

– 20-odd years teaching CS and math
– 8 years in industry, rejoined ISU in 2008
– MS in CS, PhD in Math
  • But you didn’t finish high school? Don’t tell them that!
What are we doing here

• Introduction to Object-Oriented Programming
  1.
  2.
  3.
#1: What is programming?

- “writing out a sequence of instructions for a machine to carry out?”

- Hmmm...
  - That’s a little like saying that creating a novel is accomplished by putting words together into sentences

- What about the *thinking* where you decide what the instructions should *do*?

- But, what does it mean to “design” software?
Designing software

• Suppose you have a few hundred lines of instructions
  – Tic-tac-toe game, print loan table, sort list of names...
  – Well, this is probably just a “program”

• Applications like Word or Firefox may involve a million lines of code
  – Too complex for one person to understand...
  – ...unless very carefully designed!
Object-oriented design

• This is where the “OO” comes in (item #2)
• Modern applications are too complex to be written as a simple sequence of instructions
• OO is a natural way of breaking down a complex system into components
  – Each component is simpler than the whole
  – You specify
    • What does each component do?
    • How do the components interact?
Analogy

• A typical car consists of approximately 30,000 parts
Analogy

• But it makes a lot more sense as a system of interacting components
• (Engine, Chassis, Steering...)
Objects

• In OO design and programming, the components are called “objects”
  – Within each component there are operations, containing instructions to execute...
  – But we understand and build an application as a system of interacting objects
Com S 227

• Emphasis of 227: designing, implementing, using objects effectively to make stuff

• It will still involve details of programming:
  – Variables and expressions for arithmetic and text
  – If-then control structures
  – Instructions for repetition or “loops”

• Will also cover OO concepts such as *inheritance* and *polymorphism*
Which brings us to #3

• This is technically an “introductory” course
• In practice, it can be difficult for someone who has had no exposure to programming!
  • Partly because we are introducing *programming* and *objects* at the same time
• Be sure you have the math background
  • Placement into Math 143 or calculus
• Consider taking Com S 127 first
  • Programming at a slower pace, without “objects”
Should I be taking this course?

• Many people with no previous programming experience do just fine
  – Good study skills and time management
  – A’s in Calculus, good at logical thinking
  – National merit scholar

• ...but many people, even with some programming experience, have a lot of trouble!
  – Last spring: 400 students, 135 dropped/failed
Why do so many people end up taking the course twice?

• First time: learn not to procrastinate
• Second time: learn about object-oriented programming
I’ve seen all this stuff before, can I skip Com S 227?

• Yes. Maybe. Not so fast.
  – Be careful what you wish for
  – 227 provides experience with more complex projects, rigor in design and testing to a specification…

• Take a look at the archive from last semester
  – Can you do homework 3 easily? How about the final exam review problems?
Testing out

• There is a test-out exam
  – Thursday 6-8 pm
  – University fee: $100

• You CANNOT take the test-out if
  – You have a grade or W for Com S 227, or
  – You took the test-out before
I don’t want to be a programmer

• What does programming have to do with computer science?
• Is programming too hard? Too easy? Too boring?
Course organization

• There are about 400 students this fall
• 2 lecture sections like this one
  – MWF at 1:10 and 3:10
• 8 hands-on lab sections
  – Two hours each week in groups of ~50
  – Opportunity to try things out where there are TAs and people around to answer questions!
Read the syllabus

• What’s a syllabus??
• See: http://www.cs.iastate.edu/~cs227/syllabus.html