CSC 221: Introduction to Programming

Fall 2011

course overview

- What did you set out to learn?
- What did you actually learn?
- Where do you go from here?
- How do you prepare for the exam?

What did we set out to learn?

recall the course goals from the syllabus:

- To develop problem solving and programming skills to enable the student to design solutions to non-trivial problems and implement those solutions in Python.
- To master the fundamental programming constructs of Python, including variables, expressions, functions, control structures, and lists.
- To build a foundation for more advanced programming techniques, including object-oriented design and the use of standard data structures (as taught in CSC 222).

student expectations included:

- basic programming skills
- learn more about computers & problem solving

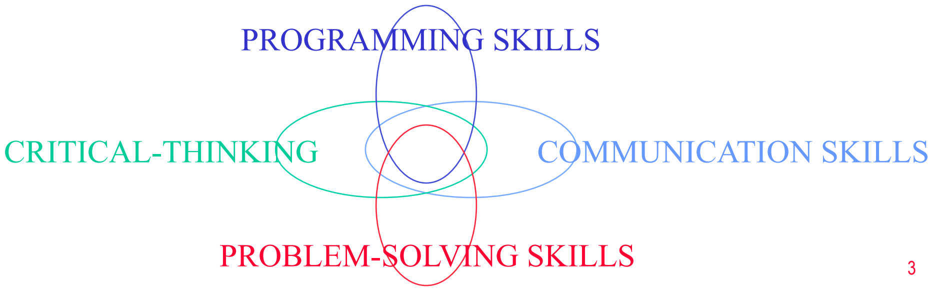
What did you actually learn?

problem-solving: the ability to take a problem, break it into manageable pieces, design and organize a step-by-step solution

programming: the ability to design and implement problem solutions in the form of programs that can be understood and executed by computers

critical-thinking: the ability to analyze and identify the important features of a problem, systematically test and evaluate solutions

communications: the ability to express ideas in a clear and precise manner, so that they could be understood by the computer (code) or another person (code & comments)



PROGRAMMING SKILLS

CRITICAL-THINKING

COMMUNICATION SKILLS

PROBLEM-SOLVING SKILLS

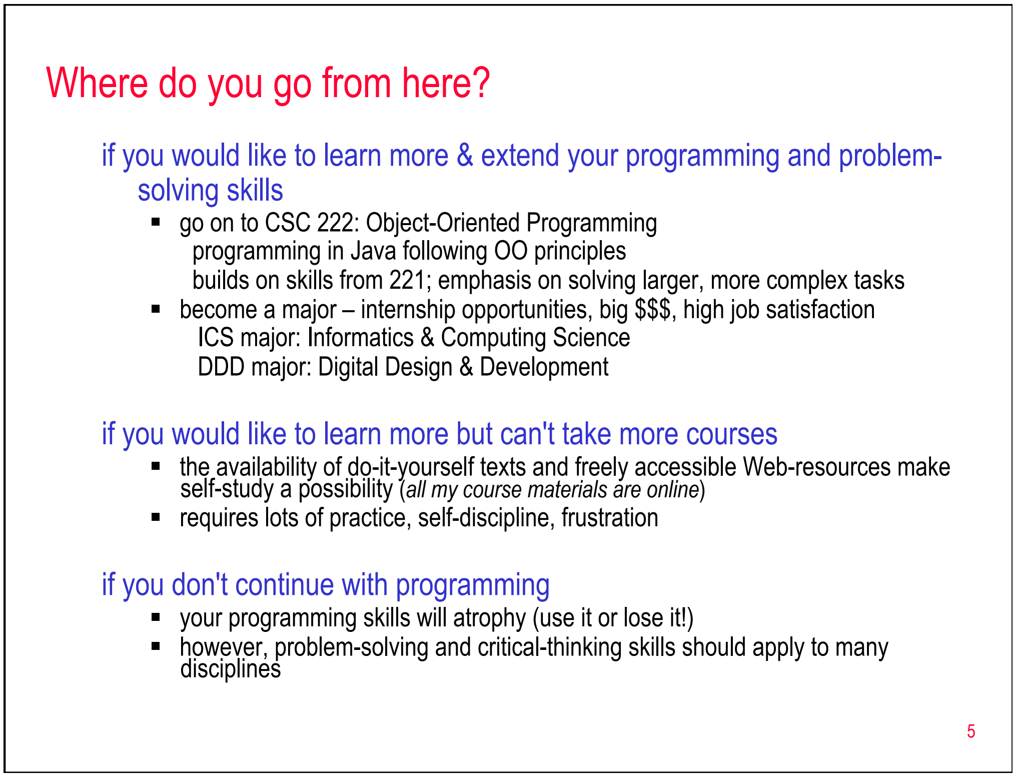
Specific Python concepts & skills

language features

- variables: data types, assignments, expressions
- functions: def, parameters, return statements
  built-in functions, user-defined functions, modules
- conditional execution: if, if-else, cascading if-elif
- repetition: while loop, for loop
- I/O: keyboard (input, raw_input, print), files (open, read, readline, write)
- sequences: len, indexing, slicing
  strings: immutable, traversal, methods (upper, strip, split, …)
  lists: mutable, traversal, methods (sort, append, remove, …)
- classes/objects: fields, methods, self, __init__, __str__

programming techniques

- conditionals for alternatives
- loops for repetition
- strings for storing/manipulating text
- lists for storing/manipulating arbitrary data
- user interaction via keyboard and file I/O
- object-oriented design: classes model real-word entities, hide details, enable reuse



Where do you go from here?

if you would like to learn more & extend your programming and problem-solving skills

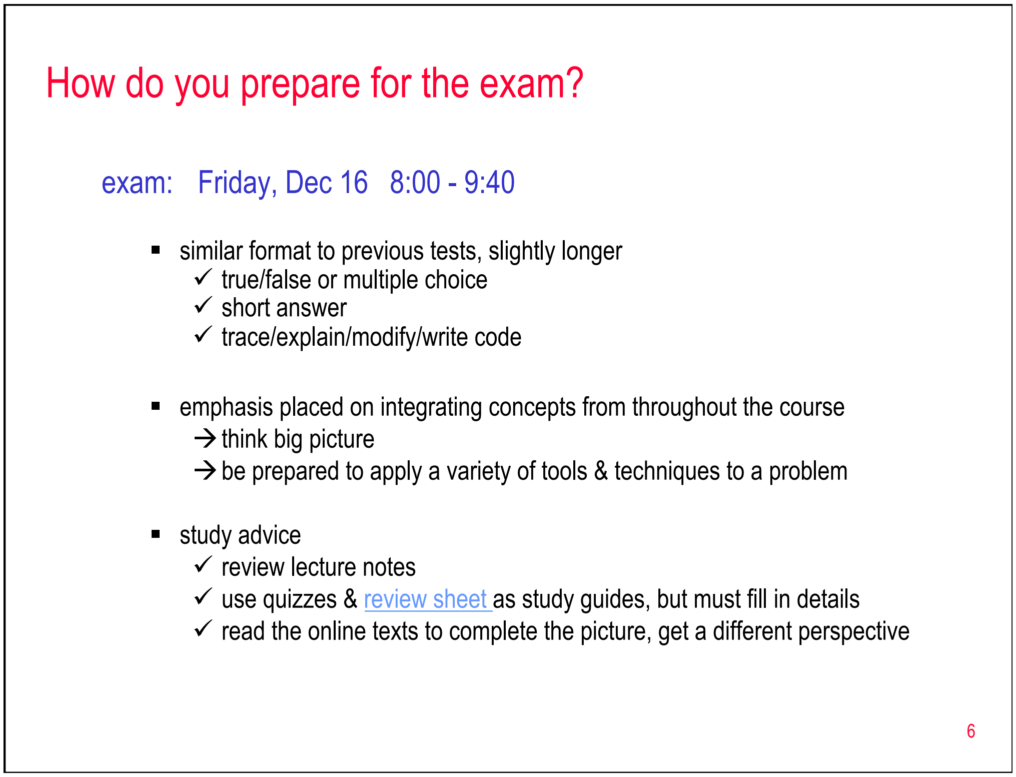
- go on to CSC 222: Object-Oriented Programming
  programming in Java following OO principles
  builds on skills from 221; emphasis on solving larger, more complex tasks
- become a major – internship opportunities, big $$$, high job satisfaction
  ICS major: Informatics & Computing Science
  DDD major: Digital Design & Development

if you would like to learn more but can't take more courses

- the availability of do-it-yourself texts and freely accessible Web-resources make self-study a possibility (*all my course materials are online*)
- requires lots of practice, self-discipline, frustration

if you don't continue with programming

- your programming skills will atrophy (use it or lose it!)
- however, problem-solving and critical-thinking skills should apply to many disciplines



How do you prepare for the exam?

exam: Friday, Dec 16 8:00 - 9:40

- similar format to previous tests, slightly longer
  ✓ true/false or multiple choice
  ✓ short answer
  ✓ trace/explain/modify/write code
- emphasis placed on integrating concepts from throughout the course
  → think big picture
  → be prepared to apply a variety of tools & techniques to a problem
- study advice
  ✓ review lecture notes
  ✓ use quizzes & review sheet as study guides, but must fill in details
  ✓ read the online texts to complete the picture, get a different perspective