CSC 221: Computer Programming I

Fall 2006

course overview

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

1

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 Java.

- To master the fundamental programming constructs of Java, including variables, expressions, classes, methods, control structures, and arrays.

- 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:

- learning more about computers
- programming skills

2

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
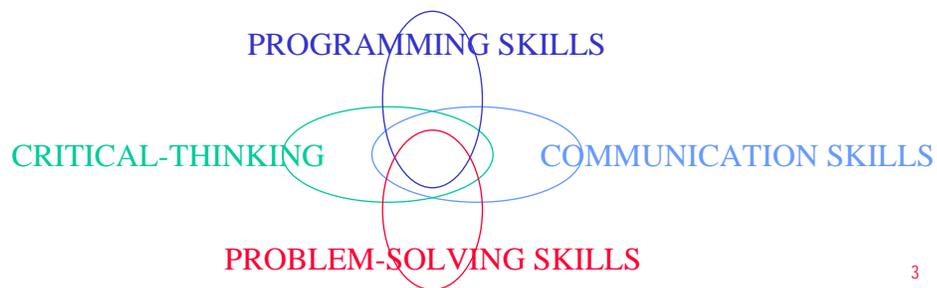
3

Specific Java concepts & skills

language features
- classes & objects
- class definitions: fields, constructors, methods
- variables: fields vs. parameters vs. local variables, primitive vs. object
- assignment statement: expressions, operators
- return statement
- if, if-else, cascading if-else
- while loop, for loop
- I/O: System.out.println, Scanner
- String, ArrayList

programming techniques
- classes model entities; methods implement behaviors; fields maintain state
- classes/methods should be cohesive; classes should be loosely coupled
- conditionals for alternatives (1-way if, 2-way if-else, multi-way cascading if)
- loops for repetition (conditional while loops, counter-driven for loops)
- counters & sums for collecting data
- String traversal/concatenation for manipulating text
- ArrayLists for storing and accessing large amounts of related data

4

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: Computer Programming II
    problem-solving & Java programming for more complex tasks
    emphasis on data structures & algorithms (as required by complex tasks)
- or CSC309: Discrete Structures
    mathematical foundations of computer science
    techniques for problem-solving and logical reasoning
- become a CS major or minor! (internship opportunities, big $$$)

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
- 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

5

How do you prepare for the exam?

exam: Monday, December 11   1:00 - 2:40

- 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 text to complete the picture, get a different perspective

6