topics: EVERYONE ADDS
  course intro
  Syllabus (First Day handout)
  software development
  writing algorithms - class exercise

handouts: syllabus, questionnaire, internet, Java style, logic diagram, mac user

=====================================================================

homework for next week : read ch. 1

complete HW 1: p.23-4, Q 1-14, p.25, Error Q1, Alg. Q 1-3, p. 27, Q1

install Java 5 from book CD-ROM or from Sun(see link on my web page)

and optionally, an IDE of your choice. I will support jGrasp, ( install from CD or go to www.eng.auburn.edu/grasp/)

=====================================================================

A. COURSE INTRO

1. course description
   this course is a first-semester programming class using Java
   • beginning programming (no programming knowledge expected)
     - program design, problem-solving
     - code structures (how to write commands)
     - data structures ( how to organize and represent data)
     - details of the Java language
   • focus and writing applications
     - web page delivery issues :
       just enough HTML to post your lab solutions

2. Syllabus includes a schedule and the first lab assignment

3. Classroom Style - combination of lecturing, computer demos, and in-class exercises, conference at end of class. Lunch break.

4. support for class on my website: demo

5. Take roll - give add codes
10-MINUTE BREAK

6. questionnaire –hand in by end of today’s class
   Any Macintosh users? Meet with me when we break for lunch.

7. Lab – ICL#1 located on 3rd floor is scheduled to reopen 9/6 – Leanna Rosenbaum, Faculty Monitor will drop by today.
   (except during first two weeks – icl1, fall05 ?) - use Hills login name + password
   (if new, read internet handout on how to log on to free school account)
   STUDENTS SHOULD ATTEND UNIX AND WINDOWS ORIENTATIONS

7. HOMEWORK for next week:
   read ch 1
   HW 1
   also, install the JDK 1.5 from book CD-ROM or from Sun website, and IDE of your choice(jGrasp, RealJ, Forte, Code Warrior, Jbuilder, etc.). You could also use Unix on hills. I will demo Unix and jGrasp in class.
B. WRITING ALGORITHMS
software development, writing algorithms, top-down stepwise refinement

1. Software Development Steps (Rethink as appropriate)
2. Algorithms
   def – an algorithm is a finite sequence of unambiguous, executable steps to solve a problem or complete a task, with a clear beginning and a clear ending.

   example – robot commands
   rt. turn(90°)
   left turn(90°)
   sit down
   step forward

   Top-Down Stepwise Refinement - write a general algorithm, then rewrite with more detail, so one step becomes many. Top-down means you start with a general description (“the top”). Stepwise means you take each step and break down into several steps. Each step becomes more detailed.

   Practice writing algorithms on everyday problems:
   demonstrate on board: cook rice

   A. First version: this is both the problem statement, and the algorithm
      1. cook rice

   B. First refinement
      1. put water in a pot
      2. put rice in a pot
      3. cook till done

   C. Second refinement
      1. measure 1 2/3 cups water
      2. put water in pot
      3. measure 1 cup of white rice
      4. add to water in pot
      5. cover pot
      6. turn on hi heat
      7. when pot lid jumps, turn heat low
      8. time 20 minutes
      9. turn off heat
      10. remove pot lid
      11. fluff rice with fork

   in groups: write 3 versions:
      do your laundry
      wash your car
      rearrange the chairs in this room for a dance