
Outline for May 8, 2009

Reading: text, §9.1–9.3

1. Overview of top-down design
 - a. Sometimes called “stepwise refinement”
 - b. Break problem into smaller pieces, plus the “glue” to hold them together
 - c. Do the glue first, with the smaller parts being stubs
 - d. Do the stubs
2. Step 1: the program, at a high level (see rps-1.txt)
 - a. Play rock, paper, scissors against the computer
 - b. Define goal, being specific
 - c. Define input
 - d. Define high-level design
3. Step 2: Data representation, smaller pieces, and main program (see rps-2.txt)
 - a. Define the routines as stubs
 - b. Do main, directly from the design
4. Step 3: First routine, who wins (see rps-3.txt)
 - a. Test it with the main (see rps-prog1.py)
5. Step 4: Second routine, computer picks (see rps-4.txt)
 - a. Be sure to print the result; useful later on, so a separate routine
 - b. Again, test it with main (see rps-prog2.py)
6. Step 5: Third routine, user picks (see rps-5.txt)
 - a. Check for errors on entry, and announce results
 - b. Again, test it with main (see rps-prog3.py)