Outline for April 17, 2014

1. Why you don’t count with floating point numbers [roundoff.py]
2. Simultaneous assignment [swap.py]
   a. Simple assignment: variable = expression
   b. Simultaneous assignment: variableA, variableB = expressionA, expressionB
3. Decision structures
   a. If statement [if0.py]
   b. Executes once, based on condition
   c. Syntax
4. Conditions
   a. Resolves to boolean value
   b. Literal booleans: True (1), False (0)
   c. Relational operators
      i. Use two arithmetic expressions connected with relational operatorsto create a boolean
      ii. Relational operators: >, >=, <, <=, ==, !=
      iii. Precedence: resolved after arithmetic operators
      iv. Connectives: and, or, not
      v. 6 > 2 + 3; "UCD" == "Sac State"
5. Two-way decisions [if1.py]
   a. if-else statements
   b. One condition, two possible code blocks
   c. else very powerful when the positive condition is easy to describe but not the negative
6. Multi-way decisions [if2.py]
   a. Can execute code based on several conditions
   b. elif (else if)
   c. else only reached if all previous conditions false
   d. Nested if statements
7. Indefinite loops: execute until a general condition is false (while)
   a. while [while.py]
   b. Contrast with for
   c. break causes program to fall out of loop (works with for too) [loop1.py]
   d. continue causes program to start loop over immediately (works with for too) [loop1.py]
8. Definite loops: execute a specific (definite) number of times (for)
   a. General form: for i in iterator
   b. Iterator is either list or something that generates a list
   c. Very common form: for i in range(1, 10)
9. range() in detail [for.py]
   a. range(10) gives 0 1 2 3 4 5 6 7 8 9
   b. range(3, 10) gives 3 4 5 6 7 8 9
   c. range(2, 10, 3) gives 2 5 8
   d. range(10, 2, -3) gives 10 7 4
10. Program: counting to 10 [toten.py]
11. Program: sum the first 10 squares [sumsq.py]