Topics for today

- Reading for this week...
  - Chapters 3 & 4: Computing with Numbers & Strings
  - Chapter 7.1–7.3: If statement
- Reminder
  - Course notes available at the course website
  - Lab work is posted after the lab; take a look at it
  - Decide how you will be taking your files with you from the lab
- Topics for today
  - Dealing with numbers and the math library
  - Strings
  - If-statement

Summary: What Can a Program Do?

- Compute “things” (using numbers, strings)
- Name “things” (using variables)
- Put sequences of statements into functions (name functions)
  - Functions can be parameterized and return values
- Import modules written by others or by you
- Input (e.g., `x = input("Enter a number: ")`)
- Output (e.g., `print "x and y are", x, y`)
- Repeat sequences of statements (for loop using range; more loop constructs later))
- Make decisions based on a condition being true (if-statement)
Clicker Question 1
What can a program not do?
A. Compute
B. Name
C. Input/output
D. Repeat
E. Randomize

Clicker Question 2
In Python, which expression does not equal 8?
A. 2 ^ 3
B. 2 * 4
C. 2 ** 3
D. 4 + 4
E. 10 - 2
Clicker Question 3

What is printed?

```
a = 3
b = 2*a / 4
a = “three”
print a, b
```

A. three 1.5
B. 3 1
C. three 1
D. 3, 1.5
E. three, 1

Problem Solving/Programming Tips

- Use variables with descriptive unit names
- Don’t re-compute quantities when they are already available
- Learn by playing
  - Don’t try to solve the entire problem on paper first
  - Build and test you program small pieces
  - Understand and internalize the tools
- Problems?
  - Start early, take breaks, seek help (TA, office hours).
Computing with Numbers

Two fundamental number representations

- **Integer**
  - Whole number (-, 0, +)
  - Represented precisely, within limits
  - Integer division truncates (3/2 is 1)
- **Floating point**
  - Real numbers (i.e., with decimals)
  - Approximate representation, but with high accuracy
  - Small errors can accumulate
    ```python
    for i in range(30000):
        x = x + 1.0/3
    ```

Reminder on Data Types

- Python tracks both the value and type of data
  - `int` with value 3
  - `str` with value “hello”
  - `float` with value 1.5
  - `list` with value `[0, 2, 5]`
- The type of a value affects the way operators behave
  - `int + int` (arithmetic addition)
  - `str + str` (string concatenation)
- Variables and other “containers” can hold a value of any type
Clicker Question 4

Which binary value below represents the decimal number 5?
(Participation points only; no penalty for wrong answer or “E”)
A. 01001
B. 0101
C. 1010
D. 000111
E. I don’t know

A Quick Primer on Binary

- In decimal representation, each digit is a factor times a power of 10…
  \[ 123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 \]
- In binary representation, each digit (“bit”) is a factor times a power of two…
  \[ 101 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \]
- What about negative numbers?
  - Decimal convention: special symbol (+ or -)
  - Binary computer convention…
    - Numbers stored in fixed-length “word” of binary bits (typically 32)
    - High order (“leftmost”) bit indicates sign (0 for +, 1 for –)
Binary Representation

- Integers stored in binary format
- Fixed length: 32 bits used for “standard” integers
  - \(2^{32} = 4,294,967,296\) combinations
  - Range: 0 to 4,294,967,295 (\(2^{32}-1\))
- But, we need sign bit to represent + and –; so only 31 bits available
  - Range: 0 to 2,147,483,647 (\(2^{31}-1\))
  - Negative range: -1 to -2,147,483,648 (\(2^{31}\))
- Remember when you work on the audio project:
  - Sound files use 16-bit integers

Long Integers

- Python supports integers that use more than 32 bits
- Long integers use “L” suffix…
  - 4294967296L
- Longs, like floats, are “contagious” in expressions, for example…
  - int + long has a long result (even if it “fits” in an integer)
  - int + float has float result (even if an exact integer)
  - \(2^{30}\)
  - \(2^{31}\)
  - \(2^{30} - 1 + 2^{30}\)
  - \(-(2^{30} - 1 + 2^{30}) - 1\)
Type Conversions

- Implicit: 3.0/2, 4L+2
- Explicit:
  - float(3)/2
  - float(x)/y
- Other explicit conversions...
  - int(4.5)
  - long(4)
  - str(25)
  - round(4.5)
  - round(-0.5)

Math Library

- Not available in Python by default, must import...
  - import math
  - math.sqrt(12)
- Or...
  - from math import sqrt
  - sqrt(12)
  - from math import *
- Helps to be consistent in what import format to use
- A related, relevant library: random
Useful Math Library Functions

- \( \sin(x) \)
- \( \cos(x) \)
- \( \tan(x) \)
- \( \arcsin(x) \)
- \( \arccos(x) \)
- \( \arctan(x) \)
- \( \log(x) \) (is base e)
- \( \log_{10}(x) \)
- \( \log(x,2) \) (base 2)
- \( \exp(x) \) (returns \( e^x \))
- \( \text{ceil}(x) \)
- \( \text{floor}(x) \)

And constants…

- \( \pi \)
- \( e \)

see tutorial/manual for more ….

Demo: \( \text{ceil} \), \( \text{floor} \), \( \text{round} \), …

Clicker Question 5

Which value is not equal to the others?

A. \( \text{ceil}(3.5) \)
B. \( \text{floor}(4.8) \)
C. \( \text{round}(3.6) \)
D. \( \text{ceil}(3.1) \)
E. \( \text{floor}(3.9) \)
Decision Structure: If-then-else

- Not covered until Chapter 7 in Zelle, but useful earlier
- Allow different sequences of statements to be executed depending on different cases

```python
if temp < 30:
    print "It is cold!"
if temp >= 90:
    print "It is hot"
    print "Drink more water"
```

Comparison Operators:
<, <=, >, >=, ==, !=

See page 203 of Zelle

# quadratic_with_IF.py
# A program that computes the real roots of a quadratic equation.
# Using a simple if to avoid program crash

import math
def main():
    print "This program finds the real solutions to a quadratic\n"
    a, b, c = input("Please enter the coefficients a, b, c: ")
    discrim = b * b - 4 * a * c
    if discrim >= 0:
        discRoot = math.sqrt(discrim)
        root1 = (-b + discRoot) / (2 * a)
        root2 = (-b - discRoot) / (2 * a)
        print "The solutions are: ", root1, root2
main()