Announcements

- Midterm Mon Oct 22
- Any notes you want (programs!)
- Textbook
- No calculators
- New TA!
- Prof. Amenta in lab hours Wd's

Interest on a debt

- When you are paying interest, compound interest is a bad thing!
- Say you owe $8000, at an interest rate of 15%, and you pay it off by paying $200 a month…
- Current program should not crash, on any input.
- Should exit without computing interest, if given bad input. Program may not do what user wants, but does not crash.

Program Structure

```python
import helper
principal = 100.00
rateString = raw_input("Enter annual interest rate:")
goodInput = helper.isFloat(rateString)
if not goodInput:
    print "Not a valid interest rate."
else:
    annualRate = float(rateString)
    monthlyRate = annualRate / 12.0
    balance = principal
    month = 0
    while month < 12:
        balance = balance + monthlyRate / 100.0 * balance
        month = month + 1
    eir = balance - principal
    print "interest earned is", eir

raw_input("Press enter to exit.")
```

Clean Up

- While writing a program, include lots of print statements
- When you're done, cut most of them out.

Pretty-printing floats

- In this program, fine to just round off floats to ints while printing:
  print "Balance =", int(balance)
- It would nice to be able to print two decimal places:
  $35.78
  $0.56
Lots of ways to do it

- Python has lots of formatting functions for printing floats.
- Python has lots of functions for playing around with strings.
- Let’s look at some string functions.

While loop on a string

- Basic idea:
  - Convert float to a string strInstrIn
  - Copy strInstrIn to another string strOut, but only keep two decimal places.
  - Converting anything to a string is done with str()

Note on functions

>>> x = 4.0/3.0
>>> x
1.3333333333333333

The value of x remains unchanged by the assignment statement:

y = str(x)

x is the argument of the function str(). Functions do not (should not) change their arguments.

Copy strInstrIn to strOut

- Copy one digit at a time.
- Getting one character out of a string:

```python
>>> s = "cow"

>>> s[0]
'c'

>>> s[1]
'o'

>>> s[2]
'w'
```

This is called indexing.

Copy string using while loop

- How do we know when we get to the end of a string?
- Length function len()

```python
s = "cow"
y = len(s) # y is now 3
```

- Argument (input) is a string
- Value (result, output) is an integer
- Index should be < len(s)

```python
i = 0
strOut = ""
while i < len(strInstrIn):
    char = strInstrIn[i]
    strOut = strOut + char
    print strOut
    i = i + 1
```
Now detect decimal point
- What to do with it? Lots of possibilities.
- Make new variable `end` to hold index at which copying should end.
- Decide what it's real value should be when we see the decimal point.

Still crashes!
- Crashes if not enough decimal places in `strIn`.
- How do we detect this?
- What do we do about it?