

CS150 - Fraction class

Note Title

2/22/2012

Announcements

- HW2 grades were emailed last night
- Next HW will be out in 1 week
- Friday - in class review
- Monday - midterm 1

A Note on Working Together

- On individual submissions, must work alone.
- Never look at someone else's code.

Functions versus methods (in a class)

- Both use same format:
def funcName(inputs)
[.]

- Outside of classes, no self variable

- In script, class methods come after
variable w/ dot:
mylist.sort()
if not in class:
print pairSum([12, 10, 16, 2], 15)

Operator overloading

pt * pt2



pt.--mul--(pt2)

With `--str--` and `--add--`, we are adapting already defined functions to work for our class.

What about multiplication?

2 versions :

$$(2, 3) * (5, 1) = 2 \cdot 5 + 3 \cdot 1 = 13$$

$$(2, 4) * 5 = (2 * 5, 4 * 5) = (10, 20)$$

$$5 * (1, 1)$$

Polymorphisms

Ability of program to behave differently depending on context.

For `--mul--`, we'll use `isinstance` to detect if the input is numeric or a point.

Can also code `--rmul--` to handle
3* Point(2,3)
↑
self

A new class :

$$\frac{-2}{-1} = 2$$

$$\frac{8}{6} = \frac{4}{3}$$

Python supports int and float, but not rationals.

What are rationals? *Fraction*

$$\frac{x}{y} \text{ where } x \text{ \& } y \text{ are integers}$$

Why useful? (What do we lose in floats?)

irrational

$$\frac{1}{3} = .3333\dots$$

Fraction class

$$\frac{2}{4} \rightarrow \frac{-1}{2}$$

- We'll store fractions in reduced form.
- Our class will be immutable.

Methods:

- __add__

- __mul__

- __div__

- __sub__

- __str__

- __float__

if $x < y$
↓
__lt__