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:
  ```python
def functionName(inputs):
  ...
  ```

- Outside of classes, no self variable

- In script, class methods come after
  ```python
  variable = dot
  myList.sort()
  if not in class:
  print pairSum([12,10,16,22,15])
  ```
Operator overloading

With `__str__` and `__add__`, we are adapting already defined functions to work for our class.

What about multiplication?

2 versions:

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

\((2, 4) \ast 5 = (2 \cdot 5, 4 \cdot 5) = (10, 20)\)

5 \ast (1, 1)
Polymorphism

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)
```
A new class:

\[-\frac{2}{1} = 2 \quad \frac{8}{6} = \frac{4}{3}\]

Python supports `int` and `float`, but not rationals.

What are rationals? Fraction

\(\frac{x}{y}\) where \(x\) and \(y\) are integers

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

Irrationals

\(\frac{1}{3} = .33333\ldots\)
Fraction class

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

Methods:

- __add__
- __mul__
- __div__
- __sub__
- __str__
- __float__

\[
\frac{2}{4} \rightarrow \frac{-1}{2}
\]