



CSE341: Programming Languages

Lecture 23

OO vs. Functional Decomposition; Adding Operations & Variants; Double-Dispatch

Dan Grossman

Fall 2011

Breaking things down

- In functional (and procedural) programming, break programs down into *functions that perform some operation*
- In object-oriented programming, break programs down into *classes that give behavior to some kind of data*

This lecture:

- These two forms of *decomposition* are *so exactly opposite* that they are two ways of looking at the same “matrix”
- Which form is “better” is somewhat personal taste, but also depends on *how you expect to change/extend software*
- For some operations over two (multiple) arguments, functions and pattern-matching are straightforward, but with OOP we can do it with *double dispatch* (multiple dispatch)

The expression example

Well-known and compelling example of a common *pattern*:

- Expressions for a small language
- Different variants of expressions: ints, additions, negations, ...
- Different operations to perform: eval, toString, hasZero, ...

Leads to a matrix (2D-grid) of variants and operations

- Implementation will involve deciding what “should happen” for each entry in the grid *regardless of the PL*

	eval	toString	hasZero	...
Int				
Add				
Negate				
...				