6 Overloading

Thursday, October 1, 2020 8:28 AM

* Start reading § 6.1 from Scott
* Lab2 is posted, due 10/5
* Exam 1 is next thursday 10/7

- static in Java
- operator overloading

$$\underset{\substack{\text{int}\\ \text{double}}}{\underline{a}} + \underset{\substack{\text{int}\\ \text{double}}}{\underline{b}}$$

→ user-defined operator overloading

```
class Polynomial {
    add(Polynomial p) {
    }
}
```

```
Polynomial p1, p2;
p1.add(p2);
```

C++
Python

overload + operator.

```
operator+ (Polynomial p1, Polynomial p2)
{
}
```

increases a PL's expressivity.

```
p1 = p1 + p2;
```

- function overloading

```
void f(int a);
void f(int a, int b);
void f(int a, b, c);
```

function signature

```
void f(int);
void f(int, int);
void f(int, int, int);
```

```
f(x, y);
```

default parameters

```
c = max(a, b);
```

```
int max(int, int);
int max(int,int);
double max(double,double);
```

6 Polymorphism

Monday, September 28, 2020 9:00 AM

Polymorphism:
Applicability to objects or expressions of multiple types.

A single interface to objects of different types.

Polymorphism itself has many forms!!! :-)

```
e.g. c= max (3, 8);
     c= max (3.14, Math.sqrt(5.5));
     c= max ("Hello", "Ni hao");
```

Python

```
def max (a, b):
    if a > b:
        return a
    else:
        return b
```

>, <, ... are defined for all primitive types

Most functional PLs allow polymorphism.

Java

```
system.out. println(a);
```

5 →

"_" →

toString() →

6 First-class Values

Monday, September 28, 2020 9:00 AM

1. can be assigned to a variable $a = 5;$
2. can be passed as a parameter $f(5)$
3. can be returned as a value of a function $f(\_) \to 5$
4. can be included in other data structures

Q: What about functions?

Functions are <u>not</u> FC objects in many PLs

6 First-class Values in Python, *contd.*

Monday, September 28, 2020   9:00 AM

Higher-order functions $h = f(g(x))$

$\text{map}(f, L1, L2, \ldots) = [f(L1(0), \ldots LN(0)),$
$f(L1(1), \ldots LN(1)),$
$\vdots$
$]$

$\text{map}(f, l)$
$\longrightarrow [f(l[0]), f(l[1]), \ldots, f(l(n-1))]$

$l1 = [1, 2, 3, 4]$

$\text{map}(\text{double}, l1)$
$\longrightarrow [2, 4, 6, 8]$

```
def power(x, y):
    return x**y
```

$\equiv x^y$

$l2 = [2, 3, 4, 5]$ $l1 = [1, 2, 3, 4]$

$\text{map}(\text{power}, l1, l_2)$
$\longrightarrow [1^2, 2^3, 3^4, 4^5]$

filter(f, l) $\text{even}(n) \longrightarrow T$ if n is even
F o/w

$l1 = [2, 3, 4, 5, 6, 7]$

$\text{filter}(\text{even}, l1)$
$\longrightarrow [2, 4, 6]$

$\text{range}(n) \longrightarrow [0, 1, \ldots, n-1]$

$\text{prime}(x) = T$ if x is prime
F o/w

$\text{filter}(\text{prime}, \text{range}(2, n)) \longrightarrow$

6 Anonymous Functions - Lambda Functions

Monday, September 28, 2020 9:05 AM

λ-Calculus ~ 1930's by Alonzo Church

$\lambda x . x^2$ def _ $(x) = x^2$

λ-expressions Alan Turing Turing m/c

$\underbrace{(\text{lambda } x : x**2)}_{\rightarrow 16}(4)$ $\lambda x . x^2 (4) \Rightarrow 16$

f = lambda x: x**2

f(4) → 16

```
def sq(x):
    return x**2
```

map(double, l1)

```
def double(x):
```

map(lambda x: 2*x, l)

$\boxed{\text{closures}}$