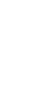


- Overloading substs. e.g. static, +, *, / user defined operator overloading (C++)
- Function Overloading

$c = \max(a, b);$

In C, will only work for values defined in $\max()$

int max (int a, int b) {



} //max()

all users of $\max()$
have to provide
int values.

Also, once defined, we cannot define another version (say, for float values):

float max (float a, float b) {



In Java, we can define both versions!!

i.e. Java allows function overloading.

- as long as we use different function signatures.

function signatures: the first line/header of a function definition.

int max (int a, int b);

float max (float a, float b);

In C, we have to use different names:

int maxint (int a, int b);

float maxfloat (float a, float b);

Polymorphism: "many forms"

has many different definitions in PL world,
making it confusing.

original: There is ONE definition for an overloaded
function (i.e. same code is executed for
parameters of different types)

e.g. In Python,

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

In Python $>$, $<$, etc
are defined for all
primitive types.

use

c = max(3, 8) # integer
c = max(3.14, math.sqrt(8/12)) # float
c = max("Hello", "Nihao")) # string

This is called Parametric Polymorphism.

Most functional PLs allow polymorphism.

Q: "Does Java have polymorphism?"

Internet says YES. But strictly speaking
it is OVERLOADING and not POLYMORPHISM.

Example Java.

abstract class shape {

{ public abstract float area(); }

} // Shape

public class Triangle extends Shape {

=
public float area() {

} // area()

=
} // Triangle

public class Square extends Shape {

=
public float area() {

} // area()

=
} // Square

public class MyProgram {

Shape T = new Triangle();

Shape S = new Square();

T.area();

S.area();

This is called subtype polymorphism (! !)

First-Class Values

e.g. int / 5

are values in a program that can be:

1. assigned to a variable e.g. $a = 5$
2. passed as a parameter of a function $f(5)$
3. returned as a value of a function return 5;
4. included in other data structures $A[i] = 5;$

Q. What about functions?

In Java: NO

In C: NO

(but you can pass a pointer
to a function)

In Python: YES

```
def double(a):  
    return 2 * a
```

double(5)

→ 10

We can also do:

- ① $x = \text{double}$
- ② $f(\text{double}, \dots)$
- ③ return double
- ④ $L[0] = \text{double}$

In Python,
functions are
also first-class objects!

Higher-order Functions: $h = f(g(x))$
requires functions as first-class objects.

e.g. Python

`map(f, L1, L2, ..., LN)`

where f is a function that
takes N arguments
 $L1, \dots, LN$ are lists

refers to

$$\begin{bmatrix} f(L1[0], L2[0], \dots, LN[0]), \\ f(L1[1], L2[1], \dots, LN[1]), \\ \vdots \\ f(L1[i], L2[i], \dots, LN[i]), \\ \vdots \\ f(L1[N-1], L2[N-1], \dots, LN[N-1]) \end{bmatrix}$$

e.g. ① $l = [1, 2, 3, 4]$

`map(double, l)`

$\rightarrow [2, 4, 6, 8]$

② def power(a, b):

return a**b

`power(2, 3)`

$\rightarrow 8 \quad \textcircled{*} = 2^3$

$l_1 = [1, 2, 3, 4]$

$l_2 = [2, 3, 4, 5]$

`map(power, l1, l2)`

$\rightarrow [1, 8, 27, 64] = [1^2, 2^3, 3^4, 4^5]$

Also, ~~f~~ filter(f, l)

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

def even(n):
 return $n \% 2 == 0$;

even(2)

→ True

even(11)

→ False

filter(even, l)

→ [2, 4, 6]

Ex: prime(n) → True, if n is prime
False o/w

range(n) → [0, 1, 2, ..., n-1]

range(a, b) → [a, a+1, ..., b]

range(1, 5) → [1, 2, 3, 4, 5]

~~try~~

filter(prime, ~~range~~ range(2, 100))

→ [all prime #'s in [2, 100]]

λ -calculus - 1930's by Alonzo Church

$$\lambda x.x^2$$

λ -expressions [Notation for defining functions]

$$\text{def } \lambda x.x^2$$

[Any computable function can be written as a λ -expression.]

function application

$$(\lambda x.x^2)(2) \Rightarrow 4$$

Essentially, you are defining a function, say f

definition $f = \lambda x.x^2$ Python def double(x):
use $f(2) \Rightarrow 4$ return $x*x$

λ -functions in a PL can be used to define in-line, anonymous functions.

e.g. Python

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

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

or, using λ -function/anonymous function

$$\text{map}(\lambda x:2*x, l)$$

Q: How to associate a stack frame??

Uses a closure: function + referencing environment
Needed when functions are first-class objects.