

CptS 355 08 Sept 2009

Note Title

9/9/2009

Last time: Ch 2 - See notes

Postscript - Programming language for
controlling printers

Post-fix notation for math expressions

1 2 +

(1+2)

1 3 2 + *

(3+2)*

3+2*

don't need () because each operation is written
right where you want it performed.

3 - 1 - 2

(3 - 1) - 2 or 3 - (1 - 2)

3 1 - 2 -

3 1 2 - -

Post script:

Programming language
uses post fix notation

numbers & other values
first use operators on

Key concepts: evaluation stack ←

dictionary stack
execution stack.

5

4 2 5
→ *
→ +
→ |
→ 3
→ 2
→ *
→ +
→ /

Elements of PS

3 3.5

numbers

(hello there)

Strings - delimiters are ()

false true

Parens not used for
arith expressions
booleans

{ 3 2 plus }

code constants

/x

the name x

when any of these is encountered
in a program the value is
pushed on the stack

{ 3 4 add } 4 5 (hi) /n dup pop count
/n
exch

5
hi
5
4
{ 3 4 add }

Operators

add, sub, mul
(dup, pop, =
stack, count, exch)

arithmetic

Stack manipulation

= : pop top element and print it
stack ~ print entire stack
w/o changing

Other Stack manip:

op index == take the op^{th} element of stack
and copy to top of stack

Program:

1 2 3 4 3 index

What is stack afterwards

n copy



Missing control flow and functions and variables.

Boolean ops

0 1 eg
4 7 lt
7 7 ge 4 7 lt and

defining

variables

ln 7 def
7
n
X 3 def
ln 4 def

table of bindings
created by def.

n	1	4
X		3

no | using variables

X 1 eg

false

Control structures

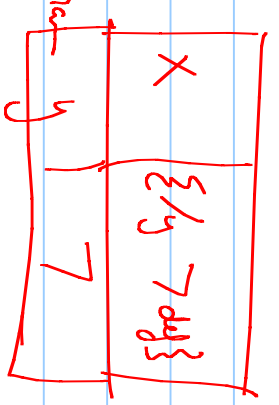
X 1 eg { } if

| if is 2-operand operation
1st operand is boolean value
2nd operand is code.

ifelse 3 - operand operator
 → bool true-code false-code if else
 x 3 eg {3 2 add} {4} if else
 bool true false

/x {1 2 7} def

when a name is bound to code
 and the name is encountered in the program
 then the code is run



3 x y add

10

