7 Control Flow - Expressions

Monday, October 5, 2020 8:40 AM

* Exam 1 is on thursday.
* No Lab this week

Refers to the flow of control of execution. Every programming language provides several mechanisms: Sequencing, selection, repetition/Iteration, procedures/functions, recursion, concurrency, exception handling, non-determinacy.

Expressions

- literals 32767, true, -4.02, "Hello"
- basic expressions a+b, a%b, a<=b, !a, a||b

-a

Infix expr: operand1 operator operand2 binary
operator operand unary

Prefix expr: operator operand1 operand2 + a b
Postfix expr: operand1 operand2 operator a b +

LISP (+ a b)
(* (+ 1 3) 2) ≡ (1+3)*2

Precedence Rules

C:
++, -- post i++, i--
++, -- pre --i, ++i
- unary
&, *
!
*, /, %
<, <=, >, >=
==, !=
&&
||

Java
!

Python
** $x**y = x^y$
- unary
*, /, %
+, -
<, <=, >, >=
not
and
or
lambda

&& } uo

|| lambda

=, +=,

7 Control Flow - Conditional Operator (?:)

Thursday, October 1, 2020 8:40 AM

Left associative $\quad a + b + c$ →

Right-associative $\quad x ** y ** z$

$(x^y)^z$ or x^{y^z}

(x**y)**z $\quad$ x** (y**z)

Conditional operator ?: $\quad$ a ? b : c

Ternary operator

```
if (a > b)
    max = a
else
    max = b
```

```
max = a > b ? a : b;
```

C, C++, Java, C#, Javascript, Swift $\quad$ max = a > b ? a : b,

Python $\quad$ max = a if a > b else b

Go $\quad$ No!

Haskell $\quad$ max = if a > b then a else b

Common Lisp $\quad$ (setf max (if (> a b) a b))

(___ (cond (> a b) a b))

Rust $\quad$ let max = if a > b {a} else {b}

Imperative PLs

Computation is carried out via expressions + assignment statements

7 Control Flow - Assignment (l-values and r-values)

Monday, October 5, 2020 8:24 AM

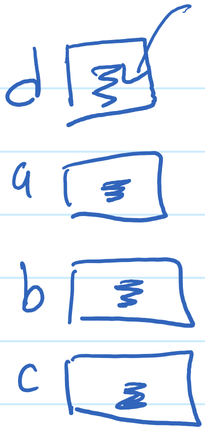
<var> = <expression>

<var> ← <epression>

<var> := <eepression>
LHS RHS

d = a;

a = b + c;



RHS: var. name refers to its value r-value

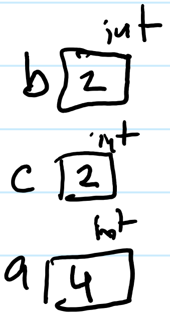
LHS: var. name refers to its location l-value

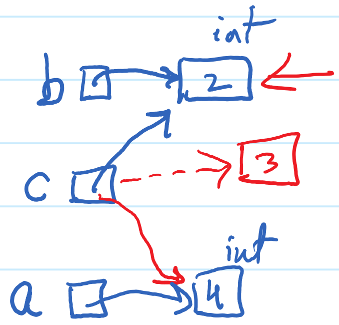
Two Models of Variables

int b = 2;

int c = b;

int a = c+b;





Value Model

Value Model: variable name is a named container for its value

Reference Model: variable name is a named reference to its value. ∴ Every name is an l-value

c = a;   c = 3;

Java   value model for all built-in types — int, long, float, double, boolean, char

reference model for all class objects — String, Arrays

C#   Class objects - reference

struct — value

Implicit Dereferencing   a = c + b | Explicit Dereferencing — deref operators.
= &a...

Implicit Dereferencing

$a = c + b$

Explicit Dereferencing

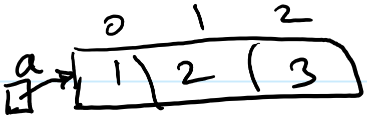
$= \&a \ldots$

7 Control Flow - Value & Reference Model for Variables

Monday, October 5, 2020   8:25 AM

Java

```
int [] a = {1,2,3};
```



```
= _ a[i] _
```

X

```
ArrayList <int> A = new ArrayList<int> ();
for (int x : a) {
    A.add(x);
}
```

```
ArrayList<Integer> A = new ArrayList<Integer>();
for (int x: a) {
    A.add( (Integer) x);  |  A.add(new Integer(x));
```

Boxing

```
A.add(x);    // is OK!!
```

Orthonality: Design goal for all PLs.

Any feature can be used in combination with any other feature.

X86

ARM ↑ orthogonal

not at all orthogonal

Addressing Modes

→ ADD	A, B, C	Direct	ADDD	A, B, C
→ ADD	@A, @B, @C	Indirect	ADDI	A, B, C

Instruction Set

small | HUGE!

RISC Processor (Reduced Instruction Set Computer) | CISC processor small

RISC Processor [Reduced Instruction Set Computer)

Huge :

CISC processor

↳ Complex