

Lecture 21:

- MyPL VM Instruction Set (cont)

Announcements:

- HW-4 due Wed
- Proj Part 2 due in 2 weeks (examples, test cases, initial design/plan)

MyPL VM Instruction Set

(3) MyPL VM Instruction Set (high level)

... see OpCode.java

Note: OP(A) says A is supplied directly to the OP instruction

- instructions take inputs directly and/or from the operand stack
- difference is what can be provided statically (directly)
- ... versus dynamically to instruction

(a) Literals and variables

PUSH(A)	push value A onto the operand stack
POP()	pop value off of the stack (remove value)
STORE(A)	pop x , store x at memory address A (a list index)
LOAD(A)	fetch x at memory address A , push x on to operand stack

MyPL VM Instruction Set

(b) Arithmetic operations ... type-based (e.g., int division)

ADD()	pop x , pop y , push $(y + x)$ on to operand stack
SUB()	pop x , pop y , push $(y - x)$ on to operand stack
MUL()	pop x , pop y , push $(y \times x)$ on to operand stack
DIV()	pop x , pop y , push $(y \div x)$ on to operand stack

(c) Logical operators

AND()	pop bool x , pop bool y , push $(y \ \&\ x)$
OR()	pop bool x , pop bool y , push $(y \ \mid\mid x)$
NOT()	pop bool x , push $(!y)$

MyPL VM Instruction Set

(d) Relational (comparison) operators

CMPLT()	pop x , pop y , push $(y < x)$
CMPLE()	pop x , pop y , push $(y \leq x)$
CMPEQ()	pop x , pop y , push $(y == x)$
CMPNE()	pop x , pop y , push $(y != x)$

(e) Jumps ... for if, while, etc.

JMP(A)	jump to instruction A (int index into instruction list)
JMPF(A)	pop x , if x is false jump to instruction A (int index)

MyPL VM Instruction Set

Simple example: `while j < 3 { j = j + 1 } ...`

```
0: LOAD(0)      # assume j stored in variables[0]
1: PUSH(3)       # literal value for the comparison
2: CMPLT()       # true if j < 3
3: JMPF(9)        # if j >= 3, jump to instruction 9
4: LOAD(0)       # get j again
5: PUSH(1)        # for the literal value 1
6: ADD()          # compute j + 1
7: STORE(0)       # store result back into j
8: JMP(0)          # go back to start of while
9: ...             # continue on after while loop
```