# Chapter 19

Register-oriented Architecture

# Register instruction set

```
        !register           ; configure to register inst set
        ld          x       ; load x into ac
        add         y       ; add y to ac
        st          sum     ; store result in sum
        ldc         @L0     ; load ac with address of @L0
        sout                ; display string pointed to by ac
        ld          sum     ; load sum into ac
        dout                ; display decimal value
        ldc         '\n'    ; load newline into ac
        aout                ; move to next line on display
        halt
x:      dw          2
y:      dw          3
sum:    dw          0
@L0:    dw          "Sum = "
```

# Need dw's for constants

```
x = y + 5000;

        ld    y
        add   @5000
        st    x

x:      dw    0
y:      dw    0
@5000:  dw    5000
```

# enter method in symbol table

```
 1 public int enter(String s, String v, boolean b)
 2 {
 3   int index = symbol.indexOf(s);
 4   if (index >= 0)     // s already in symbol?
 5     return index;     // yes, then return its index
 6
 7   index = symbol.size();
 8   symbol.add(s);      // add symbol
 9   dwValue.add(v);     // add value
10   needsdw.add(b);     // add needsdw value
11   return index;
12 }
```

# Code for register instruction set

For x + y, we get

```
ld    x
add   y
st    @t0
```

For x*y + z, we get

```
ld    x
mult  y
st    @t1

ld    @t1
add   z
st    @t2
```

For w*x + y*z, we get

```
ld    w
mult  x
st    @t3

ld    y
mult  z
st    @t4

ld    @t3
add   @t4
st    @t5
```

# Translation grammar for R1

R1.txt