CS61A – Lab Assignment 7.1 **Kurt Meinz**
University of California, Berkeley Summer 2002

Part A: Abelson and Sussman, exercises 4.27 and 4.29.

Part B: In this lab exercise you will become familiar with the Logo programming language, for which you'll be writing an interpreter in project 4.

To begin, type `logo` at the Unix shell prompt — **not** from Scheme! You should see something like this:

```
Welcome to Berkeley Logo version 3.4
?
```

The question mark is the Logo prompt, like the `>` in Scheme. (Later, in some of the examples below, you'll see a `>` prompt from Logo, while in the middle of defining a procedure)

1. Type each of the following instruction lines and note the results. (A few of them will give error messages.) If you can't make sense of a result, ask for help.

```
print 2 + 3

print 2+3

print sum 2 3

print (sum 2 3 4 5)

print sum 2 3 4 5

2+3

print "yesterday

print "julia"

print revolution

print [blue jay way]

show [eight days a week]

show first [golden slumbers]

print first bf [she loves you]

pr first first bf [yellow submarine]

to second :stuff
output first bf :stuff
end
```

```
second "something

print second "piggies

pr second [another girl]

pr first second [carry that weight]
pr second second [i dig a pony]

to pr2nd :thing
print first bf :thing
end

pr2nd [the 1 after 909]

print first pr2nd [hey jude]

repeat 5 [print [this boy]]

if 3 = 1+1 [print [the fool on the hill]]

print ifelse 2=1+1  ~
          [second [your mother should know]] ~
          [first "help]

print ifelse 3 = 1+2 ~
          [strawberry fields forever] ~
          [penny lane]

print ifelse 4 = 1+2 ~
          ["flying] ~
          [[all you need is love]]
```

Continued on next page...

Lab Assignment 7.1 continued...

```
to greet :person
say [how are you,]
end

to say :saying
print sentence :saying :person
end

greet "ringo

show map "first [paperback writer]

show map [word first ? last ?] ~
         [lucy in the sky with diamonds]

to who :sent
foreach [pete roger john keith] "describe
end

to describe :person
print se :person :sent
end

who [sells out]

print :bass

make "bass "paul

print :bass

print bass

to bass
output [johnny cymbal]
end

print bass

print :bass

print "bass
```

```
to countdown :num
if :num=0 [print "blastoff stop]
print :num
countdown :num-1
end

countdown 5

to downup :word
print :word
if emptyp bl :word [stop]
downup bl :word
print :word
end

downup "rain

;;;; The following stuff will work
;;;; only on an X workstation:

cs

repeat 4 [forward 100 rt 90]

cs

repeat 10 [repeat 5 [fd 150 rt 144] rt 36]

cs repeat 36 [repeat 4 [fd 100 rt 90]
              setpc remainder pencolor+1 8
              rt 10]

to tree :size
if :size < 3 [stop]
fd :size/2
lt 30 tree :size*3/4 rt 30
fd :size/3
rt 45 tree :size*2/3 lt 45
fd :size/6
bk :size
end

cs pu bk 100 pd ht tree 100
```

2. Devise an example that demonstrates that Logo uses dynamic scope rather than lexical scope. Your example should involve the use of a variable that would have a different value if Logo used lexical scope. Test your code with Berkeley Logo.

3. Explain the differences and similarities among the Logo operators " (double-quote), `[ ]` (square brackets), and `:` (colon).