Homework, Week 2, part D

lightprogrammong.org

September 9, 2016

A computer's memory consists of numbered cells, billions of them. The numbers are hexadecimal numbers, and raw memory addresses look like this: 35a6bf95468a20c2. Because the memory addresses are so hard to remember, we can give individual memory cells names, like *sum* or *count*. We can then place *values* in memory cells, and access the values by calling the name of the memory cell. We call this name-value combination a *variable*.

More formally, we speak of *binding* a value to a variable. When the variable has a value, we say that the variable is *bound*. When the variable does not have a value, we say that the variable is *unbound*.

Variables that can be seen and accessed throughout the entire program are called *global* variables. Variables that can be seen only in a part of the program are called *lexical* variables, or sometimes *local* variables. Lisp has a number of ways of binding variables to values. Evaluate the following statements. Then, look up the functions and read the function definitions.

```
1   ;; global variable bindings
2   ;; we will mostly use defparameter
3   (defparameter a 2)
4   (defvar b 4)
5   (defconstant c 6)
6   (+ a b c)
7
8   ;; lexical variable bindings
9   ;; we will mostly use setf
10  (setf d 3)
11  (setq e 5)
12  (set 'f 7)
13  (+ d e f)
14
15  ;; lexical variable bindings
16  ;;    used inside functions for
17  ;;    temporary variables
18  ;; also see another version, let*
19  (let
20    ((g 10)
21     (h 20)
22     (i 30))
23    (+ g h i))
```