Andrew Clayton

CSCI330

Assignment 2 Question 1

Note: for this assignment I have both this pdf and the lisp file and its output. This pdf has every single question, even for LISP code. The LISP code also has every question, but for one or two of them it just refers back to this PDF for the answer.

EXERCISES

3.1. What does (NOT (EQUAL 3 (ABS −3))) evaluate to?

3.2. Write an expression in EVAL notation to add 8 to 12 and divide the result by 2.

3.3. You can square a number by multiplying it by itself. Write an expression in EVAL notation to add the square of 3 and the square of 4.

3.1) This evaluates to NIL. !(3 == 3).

3.2) (/ (+ 8 12) 2)

3.3) (+ (* 3 3) (* 4 4))

3.7. Define a function MILES-PER-GALLON that takes three inputs, called INITIAL-ODOMETER-READING, FINAL-ODOMETER-READING, and GALLONS-CONSUMED, and computes the number of miles traveled per gallon of gas.

(defun miles-per-gallon (init-odometer-reading fin-odometer-reading gallons-consumed) (/ (- init-odometer-reading fin-odometer-reading) gallons-consumed))

3.10. The following expressions all result in errors. Write down the type of error that occurs, explain how the error arose (for example, missing quote, quote in wrong place), and correct the expression by changing *only* the quotes.

```
(third (the quick brown fox))

(list 2 and 2 is 4)

(+ 1 '(length (list t t t t)))

(cons 'patrick (seymour marvin))

(cons 'patrick (list seymour marvin))
```

```
; 3.10
; 1
(third (the quick brown fox))
; Error: 'the' operator has too many parameters (the being interpreted as an operator)
; Argument must be quoted

; Fixed:
(third '(the quick brown fox))

; 2
(list 2 and 2 is 4)
; Error: variable AND is unbound
; Symbols as data must be quoted

; Fixed:
(list 2 'and 2 'is 4)

; 3
(+ 1 '(length (list t t t t)))
; Error: value "(LENGTH (LIST T T T T))" is not of type NUMBER
; This is trying to construct a list rather than evaluating the output of the length operation.
; If we remove the quote, then the number that length returns can be added properly.

; Fixed:
(+ 1 (length (list t t t t)))

; 4
(cons 'patrick (seymour marvin))
; Error: variable marvin is unbound
; The symbols semour and marvin do not have quotes ' before them.

; Fixed
(cons 'patrick (list `seymour `marvin))
```

3.20. Here is a mystery function:

```
(defun mystery (x)
  (list (second x) (first x)))
```

What result or error is produced by evaluating each of the following?

```
(mystery '(dancing bear))

(mystery 'dancing 'bear)

(mystery '(zowie))

(mystery (list 'first 'second))
```

3.21. What is wrong with each of the following function definitions?

```
(defun speak (x y) (list 'all 'x 'is 'y))

(defun speak (x) (y) (list 'all x 'is y))

(defun speak ((x) (y)) (list all 'x is 'y))
```

3.20)

1. (BEAR DANCING)

2. Error: Function was called with 2 arguments but wants 1.

3. (NIL ZOWIE)

4. (SECOND FIRST)

3.21) For the first function, the function body will be evaluated to symbol literals rather than the variables. This will always output that same thing, “all x is y”.

For the second function, the input parameters are invalidly specified – they must be in one list (parentheses) rather than in separate ones. This will produce a syntax error.

For the third functions, the input parameters are invalidly specified again, but in a different way. Parameter lists must contain symbols (like x and y) rather than lists, like (x) and (y). The function body also tries to reference a variable “all”, which was not passed in.

3.25. What do each of the following expressions evaluate to?

```
(list 'cons t nil)

(eval (list 'cons t nil))

(eval (eval (list 'cons t nil)))

(apply #'cons '(t nil))

(eval nil)

(list 'eval nil)

(eval (list 'eval nil))
```

1. (CONS T NIL)

2. (T)

3. error undefined function T

4. (T)

5. NIL

6. (eval nil)

7. NIL