

Cha Vue

Homework 2 CSCI 330

Lisp Text

3.1. The expression evaluates to NIL

3.2. (/ (+ 8 12) 2)

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

3.7. (defun miles-per-gallon (initial-odometer-reading final-odometer-reading gallons-consumed) (/ (- final-odometer-reading initial-odometer-reading) gallons-consumed))

3.10.

- (third (the quick brown fox)) error unassigned variables. Correction = (third '(the quick brown fox))
- (list 2 and 2 is 4) error unassigned variables. Correction (list 2 'and 2 'is 4)
- (+ 1 '(length (list t t t t))) error incorrect use of quoted expression, the compiler compiles '(length as a list and does not evaluate the length of (list t t t t)).
Correction (+ 1 (length (list t t t t)))
- (cons 'patrick (seymour marvin)) error unassigned variable (seymour marvin) is not a quoted list so the compiler evaluates seymour as a function. Correction (cons 'patrick '(seymour marvin))
- (cons 'patrick (list seymour marvin)) error unassigned variable, seymour and marvin are unquoted so compiler assumes they are variables. Correction (cons 'patrick (list 'seymour 'marvin))

3.20

- (mystery '(dancing bear)) result = (bear dancing)
- (mystery 'dancing 'bear) error, function takes one argument but there are two arguments in ('dancing and 'bear)
- (mystery '(zowie)) result = (nil zowie)
- (mystery (list 'first 'second)) result = (second first)

3.21

- `(defun speak (x y) (list 'all 'x 'is 'y))` incorrect quoting of variables, function tries to return a list but because 'x and 'y are quoted the compiler treats them as literal symbols rather than evaluating it.
- `(defun speak (x) (y) (list 'all x 'is y))` Incorrect parameter list (y) is the incorrect syntax for a function definition, the parameter list should contain all parameters in one set of paranthesis.
- `(defun speak ((x) (y)) (list all 'x is 'y))` Incorrect syntax, quoted variables and symbol usage. `((x) (y))` suggests nested lists, function params should be `(x y)`. 'x and 'y are quoted which are then treated as symbols rather than evaluated variables. All and is requires quotes so lisp can treat them as variables instead of symbols.

3.25

- `(list 'cons t nil) = (cons t nil)`
- `(eval (list 'cons t nil)) = (t)`
- `(eval (eval (list 'cons t nil))) = error t is not a function, undefined function`
- `(apply #'cons '(t nil)) = (t)`
- `(eval nil) = nil`
- `(list 'eval nil) = '(eval nil)`
- `(eval (list 'eval nil)) = nil`

Sebesta Chapter 2

2. arrays and records

5. the lack of floating-point hardware in the available computers.

6. Indexing and floating-point instructions in hardware

11. character string handling, logical loop control statements and If with an optional Else clause.

14. Linguists' concerns with natural language processing, Psychologists' interest in modeling human information storage and retrieval, and Mathematicians' interest in mechanizing certain intelligent process.

15. Lisp was developed by John McCarthy and Marvin Minsky at MIT.

20. The lack of input and output statements.

25. The Department of Defense

36. Languages that do not state exactly how a result is to be computed but rather describe the necessary form and/or characteristics of the result.

37. Facts and rules.

46. The World Wide Web

51. Web browsers

52. Its syntactic appearance, the dynamic nature of its strings and arrays, and its use of dynamic typing.

57. Static semantics rule that disallows the implicit execution of more than one segment and the control expression and the case statements can be strings in C#.

59. an XML data document and an XSLT document.

60. the output is a transformed XML document, which was transformed via the instructions described in the XSLT document.