

1. From Lisp text: Questions 3.1, 3.2, 3.3 (page 81), 3.7 (pg 83) 3.10 (pg 90), 3.20, 3.21 (pg 96), and 3.25 (pg 111).

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

NIL

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

(/ (+ 8 12) (+ 1 1))

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 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 (initial-odometer-reading final-odometer-reading gallons-consumed) (/ (- final-odometer-reading initial-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))

Third's argument is missing a quote.

Correction: (third '(the quick brown fox))

(list 2 and 2 is 4)

The and variable is unbound.

Correction: (list 2 'and 2 'is 4)

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

The quote should not be there

Correction: `(+ 1 (length (list t t t t)))`

`(cons 'patrick (seymour marvin))`

Missing a quote for list (seymour marvin).

Correction:

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

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?

The result produced by evaluating `(mystery '(dancing bear))`: (BEAR DANCING)

The error produced by evaluating `(mystery 'dancing 'bear)`: invalid number of arguments

The result produced by evaluating `(mystery '(zowie))`: (NIL ZOWIE)

The result produced by evaluating `(mystery (list 'first 'second))`: (SECOND FIRST)

### 3.21

What is wrong with each of the following function definitions?

`(defun speak (x y) (list 'all 'x 'is 'y))` ⇒ variables should not be quoted.

`(defun speak (x) (y) (list 'all x 'is y))` ⇒ a function can't have two arguments: x and y

`(defun speak ((x) (y)) (list all 'x is 'y))` ⇒ Variables are parenthesized, variables are also quote.

### 3.25

What do each of the following expressions evaluate to?

`(list 'cons t nil)` ⇒ `(cons t nil)`

`(eval (list 'cons t nil))` ⇒ (T)

`(eval (eval (list 'cons t nil)))` ⇒ error by undefined function t

`(apply #'cons '(t nil))` ⇒ T

`(eval nil)` ⇒ nil

`(list 'eval nil)` ⇒ `(eval nil)`

`(eval (list 'eval nil))` ⇒ nil

## Chapter 2 Review Questions:

2. It was designed in 1945, but published in 1972
5. The slowness of interpretation was acceptable in the early 1950s to the lack of floating-point hardware in the available computers.
6. The hardware capability that first appeared in the IBM 704 computer strongly affected the evolution of programming languages was floating-point operations hardware. It prompted the development of Fortran.
7. Fortran design project began in May 1954.
8. Mathematics was the primary application area for Fortran's design.
9. Primary source of all control flow statements in Fortran I is 704 instructions.
10. Independent compilation of subroutines was the most significant feature added to Fortran I to get to Fortran II.
11. Logical loop statements and IF with an optional ELSE were added to Fortran 4
14. They were interested in AI in the late 1950s because of natural language processing.
15. LISP was developed at MIT by John McCarthy.
20. The lack of input and output statements with formatting damaged its chances for widespread use.
21. BNF was the language designed to describe the syntax of ALGOL 60.
22. FLOW-MATIC was the programming language that COBOL was based on.
23. 1959 was the year COBOL design process began.
24. Hierarchical data structures (records) appeared in COBOL origin with Plankalkul
25. (DOD) Department of Defense was the organization that was most responsible for the early success of COBOL.
36. NP Language is concerned with the what not how. Non-proc languages are those languages where you specify what conditions the answer should satisfy, but not how to obtain it.
37. Facts and rules are the two kinds of statements that populate a Prolog database.
46. The first application for Java was Embedded consumer electronic devices, such as toasters, microwaves ovens, and interactive TV systems.
51. Web applications is the application area that Javascript is most widely used.
52. Both are scripting languages that can be embedded into HTML, designed for web applications.
57. The C Switch does not provide implicit branches at the end of its code segments. C# has a static semantics rule that disallows the implicit execution of more than one segment. The rule is that every selectable segment must end with an explicit unconditional branch statement; either a break or a goto.
59. An XSLT processor is a program that takes as input an XML data document and an XSLT document
60. An output document that can be another XML document, or HTML document or plain text

2. How to submit: Create a folder named Hwork2 in your GitHub repository, and work in that folder. For Question 1, first figure out the answers (on paper), and verify them using sbcl. Then start a script session in your codespace using the command `script Q1`. Then, start sbcl inside the script session and give the LISP commands that produce the required answers. Remember to commit after you exit the script session. For Question 2, create a pdf file, named Q2. containing your answers to all the questions. Upload the two files Q1 (script file) and Q2 to your D2L dropbox