

CS 3342 – Lab 1 (100 points)
LISP Programming – CLISP (Common Lisp)
Due Date: September 11, 2022 (Sunday) - 11:59pm

In order to complete this lab, you need to install CLISP in your computer.

1) Installing in Mac:

<https://MacPorts.org/install.php>

Quickstart

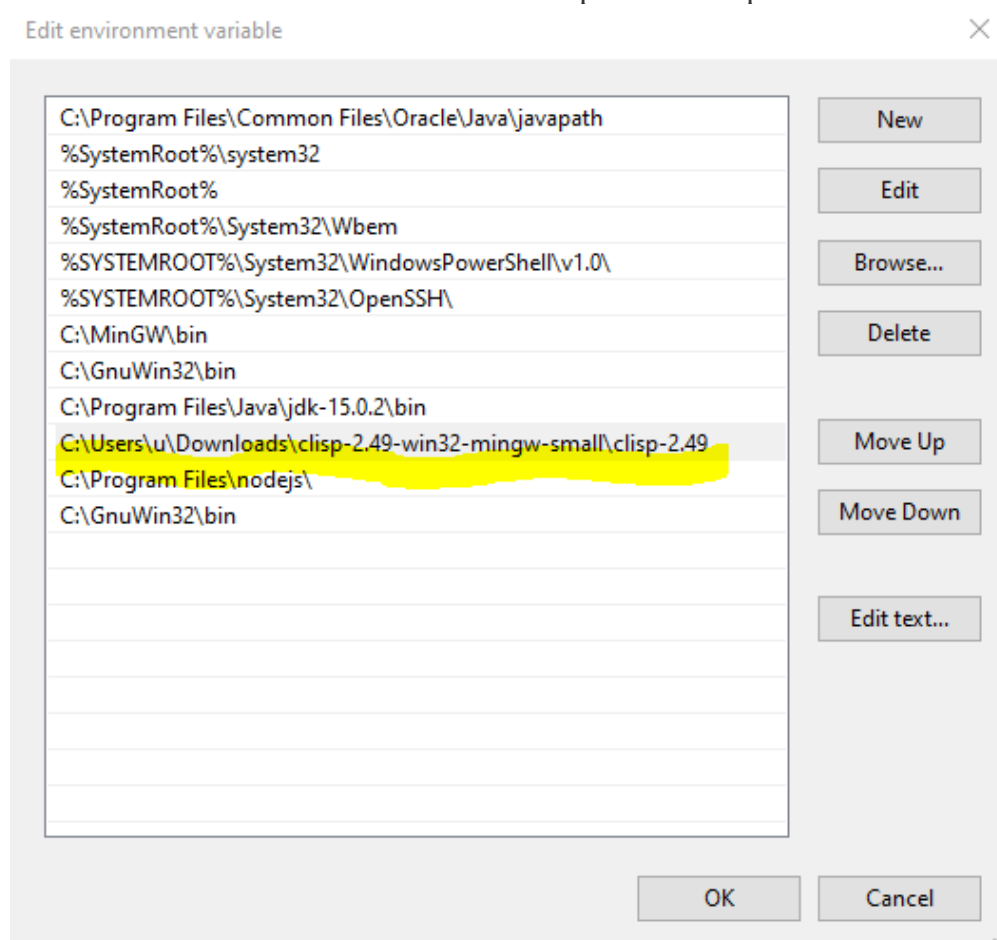
1. Install [Xcode and the Xcode Command Line Tools](#)
2. Agree to Xcode license in Terminal: `sudo xcodebuild -license`
3. Install MacPorts for your version of the Mac operating system:
 - o [macOS Catalina v10.15](#)
 - o [macOS Mojave v10.14](#)
 - o [macOS High Sierra v10.13](#)
 - o [macOS Sierra v10.12](#)
 - o [Older OS? See here.](#)

goto your terminal:

`sudo port install clisp`

2) Installing in Windows

1. Goto "<https://sourceforge.net/projects/clisp/files/clisp/2.49/>"
2. Download "clisp-2.49-win32-mingw-small.zip"
3. Unzip the above file to the location you prefer
4. Edit the Environment Variables to set the path for "clisp.exe" in 3.



3) To execute the clisp.exe program file:

Open your command prompt or terminal. Navigate to the directory where your lisp program (.lisp) is stored. Type >clisp your_file.lisp

Go through the notes in "Learning_CLISP_Student.pdf" to learn more about clisp.

Follow the instructions below to finish the tasks.

1. Hello World/Your Name

```
;;; 1. Hello Your Name
```

```
;;; ~% prints a newline with format
```

```
(format t "Hello your name~%")
```

Task: Write a CLISP program that runs like the following. Replace John with your name.

What is your name?

John

Hello JOHN, Welcome to CS3342!

2. Condition

```
;;; 2. Cond (condition) function
```

```
(defparameter *age* 18) ; Create variable age
```

```
(defvar *college-ready* nil)
```

```
;;; cond is like if else if else
```

```
(cond ( (>= *age* 18) ; If True do this
```

```
      (setf *college-ready* 'yes)
```

```
      (format t "Ready for College ~%"))
```

```
  ( (< *age* 18) ; Else If T do this
```

```
      (setf *college-ready* 'no)
```

```
      (format t "Not Ready for College ~%"))
```

```
)
```

Task: Modify the above to read the age from the terminal/console and run it with age 17 and 19.

e.g,

What is the age?

19

Ready for College

3. Factorial (sample code in a lecture ppt file.)

Your task: write a Lisp program to read a number from the terminal and find the factorial of that number. The following is sample run of the program. Use number 7 for your test. (You may use 'princ' and 'write' to print the output. (More on lisp output: [What's the difference between write, print, pprint, princ, and prin1?](#))

Running the program:

Enter a number please?

7

The factorial of 7 is

5040

4. Fibonacci Sequence and numbers

```
;;; Fibonacci Sequence
;;; if statement has this format
;;; (if (test-clause) (action1) (action2))
(defun fib (n)
  (if (< n 2)
      n
      (+ (fib (- n 1)) (fib (- n 2)))))
;;; 'collect' is a loop keyword to accumulate values in lists and return them.
;;; function to list n Fibonacci numbers.
;;; if n is 5, it will list 5 Fibonacci numbers
(defun fibs (??)
  (loop for i from 1 to ??
        collect (fib ??)))
```

Your task: modify the above code snippet to ask a user for the number of Fibonacci numbers he or she wants. Then print those Fibonacci numbers. You can use your own code to do this if it is easier for you.

Running the program:

```
How many Fibonacci number do you like to have?
10
The following are the 10 Fibonacci numbers.
(1 1 2 3 5 8 13 21 34 55)
```

5. Loop for, and car and cdr.

Car and cdr stand for “Contents of the Address Register” and “Contents of the Decrement Register”

```
;;; Get the 3rd item = aquaman
(format t “3rd Item = ~a ~%” (caddr ‘(superman batman aquaman flash joker)))
```

The above code only prints:

```
3rd Item = AQUAMAN
```

Your task: modify the code to print the following using ‘loop for’ and the car/cdr operation:

```
This is the list:
SUPERMAN
BATMAN
AQUAMAN
FLASH
JOKER
4th Item = FLASH
```

Note: In computer programming, CAR (car) and CDR (cdr) are primitive operations on cons cells introduced in the Lisp programming language. A cons cell is composed of two pointers; the car operation extracts the first pointer, and the cdr operation extracts the second.

Thus, the expression (car (cons x y)) evaluates to x, and (cdr (cons x y)) evaluates to y.

*** You can put all these code snippets in a .lisp file. Take screenshots of each output and label them in a **Word doc** so that it is easier for the grader to see your outputs.