CSC 1800 Organization of Programming Languages

This is a collection of functions that you might find helpful in writing the Lisp project. There is space between the functions for you to write notes for yourself/team.

LISP Function Orientation

Basic LISP Functions (more detail (and functions!) are in the Lispworks Docs)

(note: comments in LISP: any line starting with a ; is a comment line)

Predicates

ATOM

LISTP

CONSP

STRINGP

EQUAL

EQ

EQL

MEMBER ;; make sure you understand how to use the optional keyword arguments!

SUBSETP

Constructor Functions

LIST

CONS

APPEND

REMOVE [make sure you understand optional keyword args!)

MAKE-ARRAY

MAKE-HASH-TABLE

UNION

INTERSECTION

Destructive Functions ***(make sure you know how these differ from above)***

NCONC

DELETE

NINTERSECTION

NUNION

NSUBST

SORT ;; make sure you understand keyword arguments and how to pass a
;; function in as an argument.

Accessor Functions

FIRST / CAR

REST / CDR

NTH ;; generalized list accessor

AREF ;; (AREF A i) returns the value at the i'th location in A.

GETHASH ;; used with hash tables

Assignment Forms

SETQ ;; make sure you understand that SETQ is very limited

SETF ;; this is the more generic mechanism for assigning values to variables,
;; structure slots, positions in lists, etc.

DEFUN

LET/LET* ;; declares and initializes local variables.

DEFPARAMETER / DEFVAR / DEFCONST ;; each declaration is very different

Input/Output Functions

READ

READ-LINE

SPLIT-SEQUENCE

FORMAT ;;Know what these format directives mean: ~A , ~% , ~D

Control Forms

COND

IF

WHEN

PROGN

DOTIMES

DOLIST

LOOP

(LOOP for I from 10 to 100 doing …….)

(LOOP for j in '(a b c d) doing ……)

(LOOP for I in mylist collecting ….)

RETURN ;; not used as often as you might think!

Second Order Functions (Mapping a function onto elements of a list, collecting the results into another list to be returned)

MAPCAR

MAPCAN

MAPC