CS115 Final Exam Reference Sheet

List Functions

(cons v lst) constructs a list with first element v followed by list lst
(first lst) produces the first element in the nonempty list lst
(second lst) produces the second element in a list lst with at least 2 elements
(rest lst) produces the rest of the nonempty list lst
(length lst) produces the number of elements in the list lst
(empty? lst) produces true if lst is the empty list, false otherwise
(append l1 l2 ... ln) produces a list containing all the values in l1, l2, ..., ln, in that order, where n>1.
(list x1 x2 ... xn) produces a list of length n

Posn functions

(make-posn x y) creates a Posn object
(posn-x p) produces the x-value of Posn p
(posn-y p) produces the y-value of Posn p

Selected String and Character Functions

(char? x) produces true if x is a character, and false otherwise
(char=? c d) produces true if characters c and d are equal, false otherwise
(list->string loc) produces a string containing, in order, the characters in the list loc
(string? x) produces true if x is a string, and false otherwise
(string->list s) produces a list of the characters in string s
(string=? s t) produces true if the strings s and t are character-by-character the same, false otherwise
(string-append s t) produces a string containing all the characters in s followed by all the characters in t
(string-length s) produces the number of characters in string s
(substring s p1 p2) produces the string containing all the characters in s from position p1 to p2-1, inclusive
(substring s p1) produces the string containing all the characters in s from position p1 to the end of the string, inclusive

Abstract List Functions

(filter pred lst) produces a list of elements x, in lst, for which (pred x) => true
(foldr combine base lst) produces the value of (combine x1 (combine x2 (... (combine xN base) ...))) where lst contains x1, x2, ..., xN
(map f lst) produces a list the same length as lst obtained by applying f to each value in lst
Other Selected Functions

(add1 n) produces n+1
(equal? e1 e2) produces true if e1 and e2 are equal, false otherwise
(empty? lst) produces true if lst is the empty list, false otherwise
(max x y) produces x if x >= y, and y otherwise
(min x y) produces x if x <= y, and y otherwise
(negative? x) produces true if x < 0, false otherwise
(positive? x) produces true if x > 0, false otherwise
(quotient n m) produces the quotient when n is divided by m.
(remainder n m) produces the remainder when n is divided by m.
(sub1 n) produces n-1
(s symbol=? s t) produces true if symbols s and t are equal, false otherwise
(zero? x) produces true if x is 0, and false otherwise
(odd? x) produces true if x is odd, and false otherwise
(even? x) produces true if x is even, and false otherwise (0 is even)
(check-expect act exp) test passes if act and exp are equal, fails otherwise
(check-within act exp tol) test passes if |act-exp| <= tol
(number? x) Produces true if x is a number, false otherwise
(integer? x) Produces true if x is an integer, false otherwise