Solutions to Sample Quiz 1

Problem 1
11
error (arguments passed in wrong order)
36

Problem 2
Applicative order: Evaluate all subexpressions first, then apply the first to the rest. (Scheme uses this.)

Normal order: No arguments are evaluated until they are needed. Fully expand, then reduce.

In Scheme (applicative order), the following two items could be printed:
   one two plus
   two one plus
In normal-order Scheme, the following two items could be printed:
   plus one two
   plus two one

Problem 3
#f #f #t
#f #f #t
#f #f #t
#t #t #t

Problem 4
(caddr first-list)
(caadr second-list)

Problem 5
(define (merge list1 list2)
  (cond ((null? list1) list2)
        ((null? list2) list1)
        ((= (car list1) (car list2))
           (cons (car list1)
                  (merge (cdr list1) (cdr list2))))
        ((< (car list1) (car list2))
           (cons (car list1)
                  (merge (cdr list1) list2)))
        (else (cons (car list2) (merge list1 (cdr list2))))))

Time: Θ(n)
Space: Θ(n)
  n is dependent upon the size of the longer list
  Recursive process
Problem 6

\[(\text{define (apply\-twice } f)\]
\[\quad (\text{lambda (x) } (f (f x))))\]

Problem 7

\[(\text{tree\-manip test\-tree}\]
\[\quad 0\]
\[\quad (\text{lambda (x) } x)\]
\[\quad \text{car}\]
\[\quad \text{cdr}\]
\[\quad +)\]

\[(\text{tree\-manip test\-tree}\]
\[\quad \text{nil}\]
\[\quad (\text{lambda (x) } (\text{list } x))\]
\[\quad \text{car}\]
\[\quad \text{cdr}\]
\[\quad \text{append})\]

Problem 8

\[(\text{define (item\-name item)}\]
\[\quad (\text{caar item}))\]

\[(\text{define (item\-value item)}\]
\[\quad (\text{cdar item}))\]

\[(\text{define (item\-condition item)}\]
\[\quad (\text{cadr item}))\]

\[(\text{define first\-item car)}\]

\[(\text{define rest\-items cdr)}\]

\[(\text{define (total\-value item\-list)}\]
\[\quad (\text{if (null? item\-list)}\]
\[\quad \quad 0\]
\[\quad \quad (+ (\text{item\-value (first\-item item\-list)})\]
\[\quad \quad \quad (\text{total\-value (rest\-items item\-list))}))))\]

Time: $\Theta(n)$
Space: $\Theta(n)$
n is dependent upon the length of the list of items
Recursive process

Problem 9

\[(\text{define (car z)}\]
\[\quad (z \text{ 'car}))\]

\[(\text{define (cdr z)}\]
\[\quad (z \text{ 'cdr}))\]