

#1 During the evaluation of a function, the value of the parameter

- (a) is often updated recursively as evaluate proceeds.
- (b) is unbound until needed.
- (c) cannot change throughout the evaluation process.
- (d) may only be changed when doing so will not affect the value the function evaluates to.
- (e) is statically bound to a value.

#2 A Scheme lambda function that returns the cube of its first argument less the square of its second would be

- (a) (LAMBDA (x,y) (x^3 - y^2))
- (b) (LAMBDA (x,y) (x \* x \* x - y \* y))
- (c) (LAMBDA (x,y) (- ( (\* (x x x) (\* y y) )))
- (d) (LAMBDA (x,y) (- (^ x 3) (^ y 2)))
- (e) (LAMBDA (x,y) (- (\* x x x) (\* y y)))

#3 If a language treats functions the same way it treats data, then functions are said to be

- (a) anonymous.
- (b) first-class entities.
- (c) atomic.
- (d) higher order functions.
- (e) polymorphic.

#4 One of the main reasons that functional languages have not had the success that imperative languages have enjoyed is that

- (a) imperative languages are much more strongly matched to the capabilities of the underlying hardware.
- (b) functional languages seem too "foreign" to most programmers.
- (c) functional languages can only be applied to a very narrow range of problems.
- (d) functional languages lack the features that imperative languages require.
- (e) imperative languages place fewer restrictions on the programmer.

#5 Assuming n is a positive integer, what is returned by  
(define (fib n) (if (= n 1) n (\* n (fib (- n 1))))) ?

- (a) the product of the first n even integers
- (b) the sum of the first n integers
- (c) double the sum of the first n integers
- (d) the factorial of n
- (e) the nth fibonacci number

Enter the letter(s) of each answer below. You may choose multiple answers, but credit will be divided by the number of choices made.

1 C 2 E 3 B 4 A 5 D 6 A 7 B 8 A 9 E 10 D

#6 Scheme language requires that \_\_\_\_\_ functions be converted to iterative implementations.

- tail-recursive
- (b) head-recursive
- (c) recursive
- (d) counter-controlled looping
- (e) looping

#7 The EVAL function, by itself, serves as

- (a) evidence that a compiler is superfluous to functional languages.
- a LISP interpreter.
- (c) the primary means of executing iterative algorithms.
- (d) a standardizing force in the functional programming world.
- (e) a means of defining anonymous functions.

#8 A function that returns a Boolean value is known as

- a predicate function.
- (b) a relational function.
- (c) a declarative function.
- (d) a logical function.
- (e) a Boolean function.

#9 The basic data structure in LISP/Scheme/Racket is

- (a) the array
- (b) the atom
- (c) the string
- (d) the list
- the cons cell

#10 What will be the result of (cons (list (cons 1 2) 3 4) 5) ?

- (a) (((1 . 2) (3 4)) . 5)
- (b) ((1 . 2) . (3 . 4) 5)
- (c) (1 2 3 4 5)
- (((1 . 2) 3 4) . 5)
- (e) (((1 . 2) (3 . 4)) . 5)