

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.001 Structure and Interpretation of Computer Programs
Spring, 2007

Practice Problems, March 2

Type Problems (and let practice)

Dr. Kimberle Koile

For each expression or set of expressions, give the value and type of the value returned by evaluating the last expression in the set.

- | | value | type |
|--|-------|------|
| 1. ((lambda (x)
(+ x y))
7) | | |
| 2. ((lambda (x)
(let ((y 4))
(+ x y)))
7) | | |
| 3. (lambda (x)
(x 4 5)
) | | |
| 4. (lambda (a b c)
(+ a b)) | | |
| 5. (lambda (x y)
(lambda (x)
(y x))) | | |
| 6. (((lambda (x y)
(lambda (z)
(x y z))
)
+ 2)
4) | | |

value

type

7. ((lambda (x)
 (let ((a 1)
 (b 5))
 (if x a b))
)
 (> 20 10))

8. (define x +)
 (let ((a 3))
 (list x a a))

9. (define (foo a b)
 (let ((x 6)
 (c (+ a 5)))
 (+ b x c)))

 ((lambda (x y f)
 (f x y))
 1 2 foo)

Extra problem (not to worry about now):

10. (let ((a 10)
 (b 2))
 (let ((c (+ a b)))
 (* a c)))

Note: The second let is needed because the value of a variable is not bound until the entire list of variable-value pairs is evaluated. In this example, the value of a or b can't be used in defining c in the first let's list of variables.