Clicker Qs for November 5
What is the scope of variable j in this Java program?

public static void main(String[] args) {
    int i;
    i = 1;
    while (i < 10) {
        int j;
        j = i;
        System.out.write(i);
        i += 1;
    }
    System.out.write(i+j);
}

A. Just the while loop.
B. The while loop and the write statement that follows it.
C. The entire function main().
D. This is an error. In Java you can only declare variables at the start of a function.
Answer: A: Just the While loop
How will we evaluate something like
   ( (lambda (x) (+ x y)) 23)
in some environment ENV?

A. You can't evaluate this expression because there is no value for y.
B. Evaluate the body of the lambda expression, (+ x y), in ENV.
C. Change the binding of x in ENV to 23, then evaluate (+ x y) in ENV.
D. Extend ENV with a binding of x to 23, then evaluate (+ x y) in this extended environment.
Answer: D. Extend ENV with a binding of x to 23, then evaluate (+ x y) in this extended environment.
What is the value of this Scheme expression?

(let ([y 3])
  (let ([f (lambda (x) (+ x y))])
    (let ([y 17])
      (f 2)))))

A. This is an error because y is multiply defined.
B. 5 (in other words, the binding [y 3] applies in the function
C. 19 (in other words, the binding [y 17] applies in the function
D. The answer is indeterminant.
(let ([y 3])
    (let ([f (lambda (x) (+ x y))])
      (let ([y 17])
        (f 2)))))

Answer B: 5 (in other words, the binding [y 3] applies in the function
What does a lambda expression evaluate to???

A. lambda-exp
B. Don't tell me. It's that C-word. Caniption? Collaborator? Connundrum??
C. A CLOSURE. And don't you forget it.
D. sigh.
To be specific, how will we evaluate (lambda (params) body) in environment ENV?

A. (eval-exp body ENV)
B. We'll make a new datatype closure that holds the parameter list and the body and take those fields from the lambda expression.
C. We'll make a new datatype closure that holds the parameter list, the body, and ENV and take the first two from the lambda expression and ENV from the argument to eval-exp.
D. We'll make a new datatype closure that holds the PARSED parameter list, the PARSED body, and ENV.
E. None of these.
Answer E: None of these. Answer D is close, but we don't want to parse the parameter list.
Altogether, here is how we evaluate applications (f args) in environment ENV.

1. Evaluate f in ENV. If it evaluates to a prim-proc we have already handled it. Call apply-prim-proc with the evaluated f and the evaluated args.

2. If f evaluates to a closure it has 3 parts: body, params, closure-env. We evaluate args in ENV, then call eval-exp on body, using the environment we get by extending closure-env with bindings of the params to the evaluated args.