(define x 5)

(define (foo n)
  (+ x n))

(foo 1) => ______

(define (bar a)
  (define x a)
  (foo 1))

(bar 10) => ______

(define (baz b)
  (set! x b)
  (foo 1))

(baz 20) => ______

(define (blah c)
  (let ((x c))
    (foo 1)))

(blah 100)

(4/6/07  environment diagram example (solution))
example (explanation)

1. (define x 5)
2. (define (foo n)
   (+ x n))
3. (foo 1) => 6
4. (define (bar a)
   (define x a)
   (foo 1))
5. (bar 10) => ______
6. (define (baz b)
   (set! x b)
   (foo 1))
7. (baz 20) => ______
8. (define (blah c)
   (let ((x c))
     (foo 1)))
9. (blah 100) => ______

① add x to GE and set its value to 5
② add foo to GE and set its value to a double linkable
   whose code ptr points to parameter n and body (+ x n),
   and whose env ptr points to GE
③ eval (foo 1) | GE — drop a new frame E₁, link E₁’s ptr to
   foo’s defining env (GE) link foo’s param n to 1, and eval
   foo’s body in E₁ : (foo 1) | GE
   (+ x n) | E₁ look for x in E₁, then GE
   (+ 5 1) | E₂ ? 6
④ add bar to GE and set its value to a double linkable whose
   code ptr points to parameter a and body (define x a)
   (foo 1) ;
   and whose env ptr points to bar’s env (GE)
1. (define x 5)
2. (define (foo n)
   (+ x n))
3. (foo 1) => 6
4. (define (bar a)
   (define x a)
   (foo 1))
5. (bar 10) => 6
6. (define (baz b)
   (set! x b)
   (foo 1))
7. (baz 20) => ______
8. (define (blah c)
   (let ((x c))
     (foo 1)))
9. (blah 100) => ______

5. dual (clear 10)
   |GE: drop new frame E2, link it to bar's defining env
   |, link bar param of a to 10; then dual bar body in E2
5a. (define x a) |E2 adds x to the current env, E2, and links it to the value of a, which is 10
5b. (foo 1) |E2: drop new frame E3, link to foo's env*, link foo's param to 1,
      (+ x 1) |E3: look for x in E3, then GE
      (+ 5 1) |E3 => 6
*Note: Because foo was defined in GE, E3 points to GE; this behavior is lexical scoping. If E3 pointed to the environment from which foo was called (E2), this behavior would be dynamic scoping.

6. add baz to GE and set its value to a double linkable whose code ptr points to parameter b and body (set! x b)
   and whose env ptr points to baz's env (GE)
1. (define x 5)
2. (define (foo n)
   (+ x n))
3. (foo 1) => ______
4. (define (bar a)
    (define x a)
    (foo 1))
5. (bar 10) => ______
6. (define (baz b)
    (set! x b)
    (foo 1))
7. (baz 20) => ______
8. (define (blah c)
    (let ((x c))
     (foo 1)))
9. (blah 100) => ______
1. (define x 5)

2. (define (foo n)
   (+ x n))

3. (foo 1) => ______

4. (define (bar a)
   (define x a)
   (foo 1))

5. (bar 10) => ______

6. (define (baz b)
   (set! x b)
   (foo 1))

7. (baz 20) => ______

8. (define (blah c)
   (let ((x c))
   (foo 1)))

9. (blah 100) => ______