

CSPB 3155 - Reckwerdt - Principles of Programming Languages

[Dashboard](#) / [My courses](#) / [2244:CSPB 3155](#) / [Week 7: References and Garbage Collection](#) / [Quiz # 6](#)

Started on Wednesday, 3 July 2024, 3:20 PM

State Finished

Completed on Wednesday, 3 July 2024, 3:22 PM

Time taken 1 min 50 secs

Question 1

Correct

Marked out of 5.00

Let us assume the following store

Store

Location ID Value Stored

0	NumValue(10)
1	BoolValue(false)
2	Closure(x, DeRef(x) + 25.0) .

Note: for simplicity, we are using concrete syntax for expressions.
DeRef(x) sands for the abstract syntax DeRef(Ident("x")),
a + b for Plus(a,b) and
25.0 as Const(25.0)

and the environment

{ x : Reference(1), y: Reference(0), z: Reference(2) }

- (a) What is the value returned by the expression DeRef(x)? (Note: x must be read as Ident("x") in the abstract syntax)
- (b) What does the expression AssignRef(y, 20.0) evaluate to? (Note: y must be read as Ident("y") and 20.0 as Const(20.0). Also, according to our semantic choices made for assignref the assignment returns the new value that is actually written into the memory)
- (c) Which location in the store is updated when we execute the expression AssignRef(y, 20.0)?
- (d) Which Lettuce expression will cause the interpreter to call the closure in location 2 of the store with the argument 10.0?

DeRef(x) (10.0)	NumValue(10)	Reference(1)(10.0)	NumValue(20.0)	BoolValue(false)	DeRef(z) (10.0)
Location 2	Location 0	Location 1			

Your answer is correct.

The correct answer is:

Let us assume the following store

Store

Location ID Value Stored

0	NumValue(10)
1	BoolValue(false)
2	Closure(x, DeRef(x) + 25.0) .

Note: for simplicity, we are using concrete syntax for expressions.
DeRef(x) sands for the abstract syntax DeRef(Ident("x")),

a + b for Plus(a,b) and
25.0 as Const(25.0)

and the environment

{ x : Reference(1), y: Reference(0), z: Reference(2) }

(a) What is the value returned by the expression DeRef(x)? [BoolValue(false)] (Note: x must be read as Ident("x") in the abstract syntax)

(b) What does the expression AssignRef(y, 20.0) evaluate to? [NumValue(20.0)] (Note: y must be read as Ident("y") and 20.0 as Const(20.0). Also, according to our semantic choices made for assignref the assignment returns the new value that is actually written into the memory)

(c) Which location in the store is updated when we execute the expression AssignRef(y, 20.0)? [Location 0]

(d) Which Lettuce expression will cause the interpreter to call the closure in location 2 of the store with the argument 10.0? [DeRef(z) (10.0)]

Question 2

Correct

Marked out of 3.00

Recall Lettuce with implicit references.

- We use $\text{let var } x = e1 \text{ in } e2$ to assign x to a fresh reference that is initialized to the value of $e1$.
- We do not allow NewRef, DeRef or AssignRef directly in our language.
- We however allow assignments to x .

Recall the semantic rule for evaluating an identifier x that is currently a reference to store location j :

$$\frac{x \in \text{domain}(\text{env}), \text{env}(x) = \text{Reference}(j), \text{StoredValue}(\text{store}, j) = v}{\text{eval}(\text{Ident}(x), \text{env}, \text{store}) = v} \text{(implicit-ref)}$$

Which of the following expressions in Lettuce with implicit references will evaluate to a value of the form $\text{Reference}(j)$?

Select one:

- ☐ a. $\text{let var } x = \text{NewRef}(25) \text{ in } x$
- ☒ b. Impossible: there is no expression in this language that will evaluate to a Reference. Correct: since (a) we cannot explicitly create a reference and (b) every time we encounter an identifier that is mapped by the environment to a reference, we lookup that reference from the store and compute its value.
- ☐ c. $\text{let var } x = 25 \text{ in}$
 $\text{let dummy} = \text{assign}(x, 35) \text{ in}$
 $\text{let var } y = x \text{ in}$
 y
- ☐ d. $\text{let var } x = 10 \text{ in } x$

Your answer is correct.

The correct answer is: Impossible: there is no expression in this language that will evaluate to a Reference.

Question **3**

Correct

Marked out of 3.00

Consider the program

```
let rec f = function (x)
    if (x == 0)
    then 1
    else NewRef( f(x-1) )
in
    f(3)
```

Which of the following expressions yields the same value as the program above?

Select one:

- ☐ a. `NewRef(NewRef(NewRef(NewRef(1))))`
- ☐ b. `1`
- ☒ c. `NewRef(NewRef(NewRef(1)))`
- ☐ d. `NewRef(NewRef(1))`
- ☐ e. `NewRef(NewRef(NewRef(0)))`

Your answer is correct.

The correct answer is: `NewRef(NewRef(NewRef(1)))`

Question 4

Correct

Marked out of 4.00

Match each Lettuce program below with its return value when interpreted.

```

let rec f = function (x)
    if (x <= 0)
    then 1
    else
        x * f(x-1)
in
    f(3)

```

Value returned by the program above:

NumValue(6)

```

let rec f = function (x)
    if (x == 0)
    then 1
    else
        x * f(x-1)
in
    f(-3)

```

Value returned by the program above:

None - Program Runs Forever

```

let rec f = function (x) function (y)
    if (x <= 0)
    then 0
    else y + f(x-1)(y)
in
    f(10)(4)

```

Value returned by the program above:

NumValue(40)

```

let rec f = function (x)
    if (x == 0) then true
    else if (x == 1) then false
    else f(x-2)
in
    f(10)

```

Value returned by the program above:

BoolValue(true)

ErrorValue (or an exception is thrown)

BoolValue(false)

NumValue(16)

Your answer is correct.

The correct answer is: Match each Lettuce program below with its return value when interpreted.

```

let rec f = function (x)
    if (x <= 0)
    then 1
    else
        x * f(x-1)
in
    f(3)

```

Value returned by the program above: [NumValue(6)]

```
let rec f = function (x)
    if (x == 0)
    then 1
    else
        x * f(x-1)
in
    f(-3)
```

Value returned by the program above: [None - Program Runs Forever]

```
let rec f = function (x) function (y)
    if (x <= 0)
    then 0
    else y + f(x-1)(y)
in
    f(10)(4)
```

Value returned by the program above: [NumValue(40)]

```
let rec f = function (x)
    if (x == 0) then true
    else if (x == 1) then false
    else f(x-2)
in
    f(10)
```

Value returned by the program above: [BoolValue(true)]

Question 5

Correct

Marked out of 5.00

Let env be the environment

{ x : NumValue(25), f: Closure(x, Plus(Ident("z"), Ident("x")), {z: NumValue(1.0)}), z: NumValue(3.1415) }

Consider the recursive call:

```
let rec g = function (x)
  if (x <= 0)
  then 1
  else (f(x) + g(x-1)) in ...
```

Let us denote the body of the function g [if x <= 0) ... else f(x) + g(x-1)] as expression **e** and env1 be the environment
ExtendRec(env, g, x, **e**)

Select all the correct statement from the list below.

Select one or more:

- ☒ a. env1(x) = env(x) Correct
- ☐ b. env1(x) is undefined since x is the formal argument for the definition of g and is resolved only at its call site.
- ☒ c. env1(g) = Closure(x, **e**, env1) Correct
- ☐ d. env1(g) is undefined.
- ☒ e. env1(z) = NumValue(3.1415)

Your answer is correct.

The correct answers are: env1(x) = env(x), env1(g) = Closure(x, **e**, env1), env1(z) = NumValue(3.1415)

Question 6

Correct

Marked out of 3.00

Let env be an environment

$\{x : NumValue(25), y : NumValue(30), z : NumValue(40), f : Closure("x", Plus(x, Const(10)), env2)\}$.

Suppose we wish to extend env to handle a recursive function call

```
let rec f = e in ...
```

Let $env1$ denote the environment $ExtendRec(env, f, x, e)$ Which of the following values is obtained when we lookup f in $env1$?

Select one:

- ☐ a. $Closure(x, e, env)$
- ☐ b. $NumValue(v)$, wherein v is the result of evaluating e under env
- ☐ c. The lookup will return an "Unknown Identifier" error.
- ☒ d. $Closure(x, e, env1)$

Your answer is correct.

The correct answer is: $Closure(x, e, env1)$