CSPB 3155 - Reckwerdt - Principles of Programming Languages

Started on	Friday, 5 July 2024, 2:33 PM
State	Finished
Completed on	Friday, 5 July 2024, 2:46 PM
Time taken	13 mins 18 secs
Grade	**7.75** out of 10.00 (**78**%)

Question **1**

Correct

Mark 2.00 out of 2.00

What is the value computed by the following Lettuce program:

```
let x = 10  in
  let y =  let x = 10 in x + 10 in
     let z = let x = 20 in x + 10 in
              x
```

```
Just type the number in the box below.
```

Answer: 10 ✓

The correct answer is: 10

Question **2**

Correct

Mark 2.00 out of 2.00

Consider the following program in Lettuce:

```
let x = 10 in                (* Line 1 *)
let f = function (x)         (* Line 2 *)
     x  + 20                 (* Line 3 *)
in
     x  + f(x)               (* Line 4 *)
```

The comments are written between (* and *) markers.

(A) Which of the definitions does the x in line 3 refer to?

○ Line 4

● Line 2 ✓ Correct

○ It is an undefined usage

○ Line 1

Mark 2.00 out of 2.00

The correct answer is: Line 2

(B) Which of the definitions does the x in line 4 refer to?

● Line 1 ✓ Correct

○ Line 4

○ Line 2

○ It is an undefined usage

Mark 2.00 out of 2.00

The correct answer is: Line 1

Question **3**

Correct

Mark 3.00 out of 3.00

Consider the lettuce program below:

```
let y = 10 in  (* Program Loc. 0 *)
let x = (  (* Program Loc. 1 *)
        let x = 10 in  (* Program Loc. 2 *)
           let y = 20 in  (* Program Loc. 3 *)
                x + y (* Program Loc. 4 *)
         ) in
  x + y (* Program Loc. 5 *)

```

Select all the correct statements about how scoping and shadowing work for this particular program.
Selecting wrong options or failing to select all options will attract a penalty.

Select one or more:

☑ a.
The definition of y at program location 3 shadows that in location 0. ✓

☑ b.
The program can be evaluated without an error and yields a number as the result. ✓

☐ c.
The definition of y at program location 0 shadows that in location 3.

☑ d.
The expression "x+y" at Loc. 5 refers to y defined at location 0 and x at location 1. ✓

☐ e. The definition of x at location 2 shadows that at location 1.

☐ f.
The definition of x at location 1 shadows that at location 2.

Your answer is correct.

The correct answers are:
The definition of y at program location 3 shadows that in location 0.,

The program can be evaluated without an error and yields a number as the result.,

The expression "x+y" at Loc. 5 refers to y defined at location 0 and x at location 1.

Question **4**

Partially correct

Mark 0.75 out of 3.00

Consider Let Bindings in lettuce. Consider the following modified operational semantic rule that modifies how shadowing for let bindings is handled:

$$\frac{id \notin \textbf{domain}(\sigma),\ \textbf{eval}(e_1,\sigma) = v_1,\ \ v_1 \neq \textbf{error},\quad \textbf{eval}(e_2, \sigma \circ \{id \mapsto v_1\}) = v_2}{\textbf{eval}(Let(id, e_1, e_2), \sigma) = v_2}$$

The rule above says if "an identifier id does not belong to the environment σ, we evaluate e_1 and bind id to the resulting value when evaluating e_2".

$$\frac{id \in \textbf{domain}(\sigma),\ \ \textbf{eval}(e_2,\sigma) = v_2}{\textbf{eval}(Let(id, e_1, e_2), \sigma) = v_2}$$

The rule above says that if "an identifier id **does** already belong to the environment σ, we evaluate e_2 under the environment σ".

What is the value of the following program under these semantics:

```
let x = 10 in
   let x = 5 in
         x + 20
```

The remaining semantic rules are recalled here from our notes:

$$\frac{}{eval(\texttt{Const(v)}, \sigma) = v}\text{(const-rule)}$$

$$\frac{x \in \text{domain}(\sigma)}{eval(\texttt{Ident(x)}, \sigma) = \sigma(\texttt{x})}\text{(ident-ok-rule)} \qquad \frac{x \notin \text{domain}(\sigma)}{eval(\texttt{Ident(x)}, \sigma) = \textbf{error}}\text{(ident-nok-rule)}$$

Select one:

○ a. **error**

○ b. **30**

● c. **25** ☑

○ d. **false**

○ e. **true**

Your answer is partially correct.

The correct answer is: **30**