CSPB 3155 - Reckwerdt - Principles of Programming Languages

Dashboard / My courses / 2244:CSPB 3155 / Week 10: Object Oriented Programming / Spot Exam 4

Started on	Thursday, 25 July 2024, 6:58 PM
State	Finished
Completed on	Thursday, 25 July 2024, 7:14 PM
Time taken	15 mins 43 secs
Grade	**8.33** out of 10.00 (**83**%)

Question **1**

Correct

Mark 2.00 out of 2.00

Match the function signatures with those of their CPS version,

def bar(x : Int, y : Int): Int	def bar_cps[T] (x: Int, y: Int, k: Int => T) : T	✓
def bar(x : String, y : Int): String	def bar_cps[T] (x: String, y: Int, k: String => T) : T	✓
def bar(x : Int, y : String): Int	def bar_cps[T] (x: Int, y: String, k: Int => T) : T	✓

Your answer is correct.

The correct answer is:
def bar(x : Int, y : Int): Int → def bar_cps[T] (x: Int, y: Int, k: Int => T) : T,

def bar(x : String, y : Int): String → def bar_cps[T] (x: String, y: Int, k: String => T) : T,

def bar(x : Int, y : String): Int → def bar_cps[T] (x: Int, y: String, k: Int => T) : T

Question **2**

Partially correct

Mark 1.00 out of 2.00

You are given an environment env

```
{
  x -> Reference(0),
  y -> NumValue(0),
  z -> Reference(1),
  f -> Closure("x",  x + deref(z),  { z -> Reference(1)) } )
}
```

Note: read the expression x + deref(z) in the Closure as if it is written in abstract syntax " Plus(Ident("x"),...)".

The store has the following values at addresses 0, 1 and 2:

```
Addr -- Value
 0 -- NumValue(5)
 1 -- NumValue(7)
 2 -- Closure("z", z + 20 - f(10), env)
```

For each of the expressions shown below in the concrete syntax, match them to the appropriate values when evaluated under the environment and store given above.

f(x)	NumValue(12)	✗
f(y)	NumValue(7)	✓
deref(x)	NumValue(5)	✓
x + deref(y)	NumValue(12)	✗

Your answer is partially correct.

You have correctly selected 2.
The correct answer is:
f(x) → error,

f(y) → NumValue(7),

deref(x) → NumValue(5),

x + deref(y) → error

Question **3**

Correct

Mark 4.00 out of 4.00

Here is a scala function:

```
def funcy ( x: Int, y: String) : String = {
       if (x <= 0) {
                 y + x.toString
       } else {
               funcy( x - 1,  funcy ( x - 2 ,  x.toString + "," +  y ) )
      }
}
```

Fill in the missing part of the CPS transformation of the above function:

```
  def funcy_cps [T]( x : Int, y : String, k : ???1 => T) : T = {
        if ( x < = 0) {
                k ( ???2 )
        } else {
               funcy_cps ( ???3 , x.toString + "," + y,  (v) => { funcy_cps( ???4, ???5, k) } )
        }
  }
```

???1 String ✓

???2 y + x.toString ✓

???3 x - 2 ✓

???4 x - 1 ✓

???5 v ✓

Your answer is correct.

The correct answer is:
???1 → String,

???2 → y + x.toString,

???3 → x - 2,

???4 → x - 1,

???5 → v

Question **4**

Partially correct

Mark 1.33 out of 2.00

Consider the recursive function in lettuce

```
let rec f = function (x) {
                   if (x <= 1)
                   then x
                   else x + f(x - 2)

             } in
     f(20)
```

Let env denote the environment under which the subexpression f(20) is being evaluated. Select all the facts that are true about this environment.

Let *body* denote the expression of the body of the function f: "if (x<=1) ... else x + f(x-2)".

Select one or more:

☑ a. ✓

Looking up the identifier "f" in env yields Closure("x", body, env)

☐ b.

Looking up the identifier "x" in env yields Closure("x", body, env)

☐ c.

The identifier "x" is not defined in the environment env.

☑ d. ✓

env is written ExtendRec("f", "x", body, EmptyEnvironment)

Your answer is partially correct.

You have correctly selected 2.
The correct answers are:

env is written ExtendRec("f", "x", body, EmptyEnvironment)

,

Looking up the identifier "f" in env yields Closure("x", body, env)

,

The identifier "x" is not defined in the environment env.