New submission for: Quiz 7

Due: 11/17/2020 @ 04:19 PM EST

Honor Pledge

On my honor, I have neither given nor received any unauthorized aid on this quiz.

By typing your first and last name in the space provided below you are electronically signing to indicate that:

(1) You are the person who is taking this quiz.
(2) You read and understood the Honor Pledge and you agree to be bound by it.

Write your answer below:

```
Xinhao Luo
```

Clear | Use Most Recent Submission

Question 1. (2pts) Haskell's scoping discipline is

Select one:
- (•) static scoping
- ( ) dynamic scoping

Clear | Use Most Recent Submission

Question 2. (2pts) Haskell's typing discipline is

Select one:
- (•) static typing
- ( ) dynamic typing

Clear | Use Most Recent Submission

Question 3. (6pts, 2pts each line) The code below implements in Haskell (the heart of) your HW4. Fill in the blanks: the type signature of `find`, and the `Or` and `Let` case arms of `eval`.

```
type Name = String
data Expr = Var Name
          | Val Bool
          | And Expr Expr
          | Or Expr Expr
          | Not Expr
          | Let Name Expr Expr

-- Purpose: looks up variable n in binding environment env.
--    Returns first binding or throws Exception if no binding of n in env.
-- Example: find "x" [("x",True),("x",False),("y",True)] returns True
find :: _________________________________  -- YOUR CODE HERE
find n env = head [ bool | (var,bool) <- env, var == n ]

-- Purpose: evaluates expression e in binding environment env.
--    Returns the boolean value of e or throws an Exception.
-- Example: eval (Var "x") [("x",True),("x",False)] returns True
eval :: Expr -> [(Name,Bool)] -> Bool
eval e env =
   case e of
     Var n -> find n env
     Val b -> b
     And e1 e2 -> (eval e1 env) && (eval e2 env)
     Or e1 e2 -> ______________________________ -- YOUR CODE HERE
     Not e1 -> not (eval e1 env)
     Let n e1 e2 -> ______________________________ -- YOUR CODE HERE
```

Write your Haskell below:

Press TAB to indent. Press ESC to advance from answer area.

```
type Name = String
data Expr = Var Name
          | Val Bool
          | And Expr Expr
          | Or Expr Expr
          | Not Expr
          | Let Name Expr Expr

-- Purpose: looks up variable n in binding environment env.
--    Returns first binding or throws Exception if no binding of n in env.
-- Example: find "x" [("x",True),("x",False),("y",True)] returns True
find :: Name -> [(Name, Bool)] -> Bool -- YOUR CODE HERE
find n env = head [ bool | (var,bool) <- env, var == n ]

-- Purpose: evaluates expression e in binding environment env.
--    Returns the boolean value of e or throws an Exception.
-- Example: eval (Var "x") [("x",True),("x",False)] returns True
eval :: Expr -> [(Name,Bool)] -> Bool
eval e env =
   case e of
     Var n -> find n env
     Val b -> b
     And e1 e2 -> (eval e1 env) && (eval e2 env)
     Or e1 e2 -> (eval e1 env) || (eval e2 env) -- YOUR CODE HERE
     Not e1 -> not (eval e1 env)
     Let n e1 e2 -> (eval e2 ((n (eval e1 env)):env) ) -- YOUR CODE HERE
```

Clear | Use Most Recent Submission

By clicking "Submit" you are confirming that you have read, understand, and agree to follow the Academic Integrity Policy.

Submit

Select Submission Version: Version #1 GRADE THIS VERSION | Do Not Grade This Assignment

Note: This version of your assignment will be graded by the instructor/TAs and the score recorded in the gradebook.

	Autograding Total *(Without Hidden Points)*
10 / 10	**Autograding Total** *(With Hidden Points)*
	Test 1 Honor Pledge signature submission Show Details
	Test 2 Question 1 submission Show Details
	Test 3 Question 2 submission Show Details
	Test 4 Question 3 submission Show Details
2 / 2	**HIDDEN: Test 5 Question 1**
2 / 2	**HIDDEN: Test 6 Question 2**
	HIDDEN: Test 7 Compilation of student Q_03.hs and test1 *Compilation Errors and/or Warnings.*
	HIDDEN: Test 8 find
6 / 6	**HIDDEN: Test 9 Question 3**
	Test 10 Check Honor Pledge Show Details
	Test 11 Check Time Limit Show Details