CSPB 3155 - Reckwerdt - Principles of Programming Languages

Dashboard / My courses / 2244:CSPB 3155 / Week 9: Types and Type Checking / Online Quiz 8

Started on	Thursday, 18 July 2024, 9:15 PM
State	Finished
Completed on	Thursday, 18 July 2024, 9:18 PM
Time taken	2 mins 40 secs
Marks	24.00/24.00
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 4.00 out of 4.00

Match each of the following Lettuce programs with the corresponding types

```
let x: num = 25 + 15 in
              let y: bool = x >= 35 in
                   if (y) then x else x - 1
```

num

```
     let f : ( num => num ) => (num => num) = function (g: num => num) function (x: num) g( x + g(x) ) in
                let h : num => num = function (y)   2 * y in
                          f(h)
```

num => num

```
let rec f : num => num = function (x : num) f(x - 1) in f(20)
```

num

num => num | num | bool | num => bool

Your answer is correct.

Correct

Marks for this submission: 4.00/4.00.

Question **2**

Correct

Mark 5.00 out of 5.00

Select all well typed Lettuce expressions from the list below.

Select one or more:

☐ a.

```
let f: num=> num = function (x: num) x in
   if (f(true)) then f(20) else f(40)
```

☐ b.

```
let f = function (x) x >= 10 in f(25) <= 5
```

☑ c.

```
let f = function (g) function (x)  g( 1 + g(x)) in
     let h = function (x) x + 1 in
                f (h)
```

☐ d.

```
if ( 25 >= 15) then true else 45
```

☑ e.

```
let x = 10 in (x + x) >= 10
```

☑ f.

```
let rec f = function (x) x * f(x−
1) in f(25)
```

Note that the recursion is non terminating but that has no bearing on the type of function f, which is num => num

Your answer is correct.

Correct

Marks for this submission: 5.00/5.00.

Question **3**

Correct

Mark 9.00 out of 9.00

[9 points] EXPLICIT TYPE CHECKING

(A: 2 Points) Consider the following Lettuce Program

```
let f : num => (num => num) =
                     function (x :num)
                          function (y : num)
                                   x + y
in
    f(2)
```

What is the type of the (value returned by) above program? (select one)

○ num

○ Type Error

○ num => bool

● num => num

○ bool

Mark 2.00 out of 2.00

(B: 2 Points) Consider the following Lettuce Program

```
let f : num => (num => num) =
                     function (x :num)
                          function (y : num)
                                   x + y
in
      f(2)(5)
```

What is the type of the above program? (select one)

● num

○ num => bool

○ num => num

○ Type Error

○ bool

Mark 2.00 out of 2.00

(C: 2 Points) Consider the following Lettuce Program

```
  let f : (num => bool) => num => num =
     function (g :num => bool)
          function (y : num)
              g(y)
   in
      f(2)
```

What is the type of this program? (select one)

● Type Error

○ num => bool

○ num => num

○ num

○ bool

Mark 2.00 out of 2.00

(D: 3 Points) Consider the following Lettuce Program

```
let f : (num => bool) => num => num =
    function (g :num => bool)
        function (y : num)
            g(y)
in
    f (function (x: num) x > 2) (3)
```

What is the type of this program? (select one)

○ num

○ num => bool

○ num => num

○ bool

● Type Error

Mark 3.00 out of 3.00

Correct

Marks for this submission: 9.00/9.00.

Question **4**

Correct

Mark 2.00 out of 2.00

Consider the following classes in scala:

```
abstract class A (val n: Int)

class B (val n: Int) extends A

class C (val n: Int) extends A

class D (val n: Int) extends B

def foo (a: A) = { // code omitted }
```

Which of the expressions can be passed as an argument for a call to foo without resulting in an error? new D(10)

"foo" "hello" 25 new A(15)

Your answer is correct.

Correct

Marks for this submission: 2.00/2.00.

Question **5**

Correct

Mark 4.00 out of 4.00

Consider the abstract class T and classes A and B as defined below.

```
abstract class T {
   val n : Int
   def foo(n: Int): Unit
}

class A extends T { ... // contents omitted }

class B extends A { ... // contents omitted }
```

Select all facts from the list below that must be necessarily true about the members of class A and B.

Select one or more:

□ a. class B must override the members n and foo defined in A.

□ b. If class B overrides member foo then class A does not need to do so.

☑ c. class B is considered to be a subclass of T

□ d. class A can change the type of the members n and foo while overriding them from T.

☑ e. Class A must implement members n: Int and foo: Int => Unit.

☑ f. Class B may implement members n: Int and foo: Int => Unit that override the definitions inherited from A

Your answer is correct.

Correct

Marks for this submission: 4.00/4.00.