

Fall, 2007

Name: _____

COP 4020 — Programming Languages I

Test on the Declarative Model

Special Directions for this Test

This test has 9 questions and pages numbered 1 through 5.

This test is open book and notes.

If you need more space, use the back of a page. Note when you do that on the front.

Before you begin, please take a moment to look over the entire test so that you can budget your time.

Clarity is important; if your programs are sloppy and hard to read, you may lose some points. Correct syntax also makes a difference for programming questions.

When you write Oz code on this test, you may use anything we have seen in chapters 2–3 of our textbook. But unless specifically directed, you should not use imperative features (such as cells).

You are encouraged to define functions or procedures not specifically asked for if they are useful to your programming; however, if they are not in the Oz base environment, then you must write them into your test.

For Grading

Problem	Points	Score
1	15	
2	10	
3	15	
4	10	
5	10	
6	10	
7	10	
8	10	
9	10	

1. Consider the following Oz statement.

```
P = proc {$ X Q ?R}
    local Z in
        {Plus X A Z}
        if {IsNegative Z}
        then {P {Times Z X} A R}
        else R = Z
        end
    end
end
```

- (a) (10 points) Write, below, in set brackets, the entire set of the variable identifiers that occur free in above statement. For example, write $\{V, W\}$ if the variable identifiers that occur free are V and W . If there are no variable identifiers that occur free, write $\{\}$.
- (b) (5 points) Write, below, in set brackets, the entire set of the variable identifiers that occur bound in above statement. For example, write $\{V, W\}$ if the variable identifiers that occur bound are V and W . If there are no variable identifiers that occur bound, write $\{\}$.
2. (10 points) Why does a closure need to store the environment for the free variable identifiers used in a procedure? (Circle the letters of all the following that are correct. Don't circle incorrect statements. There may be zero, one, two, or more correct statements.)
- (a) Because storing the environment resolves the concept of freedom of expression in the procedure's body.
 - (b) Because it ensures that the values of the variable identifiers that occur free in the procedure are those associated with the closest textually-surrounding declarations.
 - (c) Because it allows the procedure's code to access the values of the variable identifiers that occur free in the procedure.
 - (d) Because it ensures that the variable identifiers that occur free in the procedure have the type of data associated with the closest textually-surrounding declaration.
 - (e) Because it ensures that the procedure's code uses dynamic scoping for the variable identifiers that occur free in the procedure.

3. (15 points) Desugar the following Oz code into kernel syntax by expanding all syntactic sugars. (Assume that `Sub1` and `Equal` are functions declared elsewhere.)

```
fun {G N F X}  
  if {Equal N 0}  
  then X  
  else Z={G {Sub1 N} F X} in {F Z}  
  end  
end
```

4. (a) (3 points) Which kind of type checking does Oz have: static or dynamic?

(b) (3 points) Which kind of type checking does Java have: static or dynamic?

(c) (4 points) Briefly answer the following. In a language with dynamic type checking, when are type errors caught, in general?

5. (10 points) What happens when the following code executes in Oz? Briefly explain why that happens.

```
local X in
  X = 0
  X = X + 1
  {Show X}
end
```

6. (10 points) What happens when the following code executes in Oz? Briefly explain why that happens.

```
local A X in
  A = knights(go: X reside: orlando research: languages)
  case A of
    knights(go: C reside: orlando) then {Browse first#C}
  [] knights(go: ok reside: orlando speak: languages) then {Browse second}
  [] knights(go: C reside: orlando research: D) then {Browse third#C#D}
  [] knights(go: C reside: P research: D) then {Browse fourth#C#P#D}
  else {Browse none}
  end
end
```

7. (10 points) In Oz, could one possibly think of an **if-then-else** statement as a syntactic sugar or a linguistic abstraction? Briefly explain your answer.

8. (10 points) Briefly, why does a programming language need a built-in mechanism for throwing and catching exceptions?
9. (10 points) Which of the following are correct statements about tail recursion. (Circle the letters of all the following that are correct. Don't circle incorrect statements. There may be zero, one, two, or more correct statements.)
- (a) A procedure is tail recursive only if whenever it makes a recursive call, there are no pending computations that need to be executed after such a recursive call returns.
 - (b) A procedure is tail recursive only if works on lists and only if it always makes a recursive call passing itself the tail of the list.
 - (c) A procedure is tail recursive only if it tells a story that involves repeated swearing and cursing.
 - (d) Procedures in Oz that are tail recursive can always be executed in a constant amount of time.
 - (e) Procedures in Oz that are tail recursive can be executed without the stack growing due to their recursive calls.