CSCI.4430/6969 Programming Languages Spring 2011
Programming Assignment #2

This assignment is to be done either individually or in pairs. Do not show your code to any other group and do not look at any other group's code. Do not put your code in a public directory or otherwise make it public. However, you may get help from the TAs or the instructor. You are encouraged to use the RPILMS Discussions page to post questions so that other students can also answer and see the answers.

An interpreter for a subset of Scheme in Oz

The goal of this assignment is to write an interpreter in Oz for a restricted version of Scheme. The interpreter should be able to reduce Scheme expressions that correspond to the lambda calculus in a call-by-value (applicative order) manner.

You are to use the following grammar for the lambda-calculus:

```
<expression> ::= <atom>
             ::= lambda(<atom> <expression>)
             ::= [<expression> <expression>]
```

Your interpreter is expected to take the lambda calculus expression and perform repeatedly beta reduction until no longer possible (a value expression that can no longer be beta-reduced) and then eta reduction until no longer possible.

% Once your interpreter is finished, you should be able to execute statements such as:

```
{Browse {Run [lambda(x x) y]}} % should display y
```

Hints: You may define auxiliary procedures for alpha-renaming, beta-reduction, and eta-reduction. For beta reduction, you may want to write an auxiliary procedure that substitutes all free occurrences of a variable in an expression for another expression. Be sure that the replacing expression does not include free variables that would become captured in the substitution. Remember that in call-by-value, the argument to a function is evaluated before the function is called.

10% Extra Credit:

Create a call-by-name (normal order evaluation) restricted Scheme interpreter.

See the professor if you have ideas for other extensions to this assignment and would like extra credit for implementing them.

Due Date:

Received Time	**Grade Modification**
Before Thursday, March 3rd, 11:59PM	+10%
Friday, March 4th, from 12:00AM to 11:59PM	no modification (on time)

Saturday, March 5th from 12:00AM to 11:59PM	-10%
Sunday, March 6th, 12:00AM to Monday, March 7th, 11:59PM	-25%
After Tuesday, March 8th, 12:00AM	not accepted

The assignment will be graded mostly on correctness, but code clarity / readability will also be a factor (comment, comment, comment!). **Be sure your solutions are robust**.

Submission Requirements: Your code should consist of an Oz file, plus a README file. Combine these files into a single ZIP file with your LMS user name(s) as the filename, either userid1.zip or userid1_userid2.zip. Only submit one assignment per pair (the other does not need to submit anything via RPILMS). Please submit your file via RPILMS.