For this assignment, you will again work with a partner. You will write up and turn in this assignment in pairs. Choose one of you to upload your write up to the moodle. Also, please include your PL-Detective user ids on your write-up. As always, you are welcome to discuss in larger groups. Just be sure to acknowledge those with which you discussed.

Some questions in this assignment use the PL-Detective. You will need to refer to MYSTERY grammar to answer them:

http://machine.cs.colorado.edu/~diwan/pldetective/pldsyntax.htm

The most important part of these questions is not the final answer (e.g., the kind of scoping used) but the reasoning that led to the answer. Thus, do not leave the write up to the last minute! Also, since the number of attempts you can submit to the PL-Detective (e.g., number of PRINTs that you can execute) is limited, it is worth discussing each submission to the PL-Detective with your group. We will look at submissions from both you and your partner. The limit is set high enough that you can waste about half of the attempts and still get the full score. When you submit a program to the PL-Detective, it tells you how many programs you have submitted without syntax errors so far; note that this is not directly correlated with the limit (e.g., the limit is often based on executed PRINT and one of your submissions may not do any PRINT).

Note that the PL-Detective link for each question is different. You must use the appropriate link for each question.

Exercise 1: Bookkeeping. Indicate in a sentence or two how much time you spent on this homework, how difficult you found it subjectively, and what you found to be the hardest part. Tell me something about yourself that I do not already know. Any non-empty answer will receive full credit.
Also, if your opinions have changed since the last assignment, indicate one thing you like about the class so far and one thing you would change about it.

**Exercise 2: Skill 4.3.** The textbook defines four kinds of storage bindings: static variables, stack-dynamic variables, explicit heap-dynamic variables, and implicit heap-dynamic variables. Of these categories, we will consider only the first three kinds in this question (i.e., ignore “implicit heap-dynamic variables”).

MYSTERY allows programs to declare variables associated with any block (delimited by BEGIN...END). However, declaring variables is just a matter of syntax. The kind of variables that MYSTERY actually supports (i.e., the semantics) is actually a MYSTERY. In this assignment, you will put on your detective hat and try to figure out what kinds of variables MYSTERY supports. To accomplish this you will submit programs to the PL-Detective, observe any outputs or errors, and make inferences. Based on the output you receive, you may continue the interrogation by submitting another program or decide that you have solved the MYSTERY.

1. Based on an inspection of its syntax, do you believe that MYSTERY supports explicit heap-dynamic variables? Explain and justify your answer.

2. Determine and describe the storage bindings of the variables associated with blocks in MYSTERY. Use

   [http://machine.cs.colorado.edu/~diwan/pldstorage.htm](http://machine.cs.colorado.edu/~diwan/pldstorage.htm)

   to access the PL-Detective for this question. You can use up to a total of four executed PRINT commands in your interrogation (e.g., if you submit two programs, the total number of values the two programs print out combined should not exceed four). Note that programs with syntax errors will not count towards your total since a program with a syntax error will not execute PRINT. You will lose 5% of the points for this question for each additional PRINT that you execute. The PL-Detective keeps track of all programs ever submitted by your group.

   *Hint:* your answer to part 1 should help eliminate some possibilities this part.

Give the evidence and your reasoning based on which you arrived at your conclusions. The evidence takes the form of programs you submitted and the output you received. The reasoning takes the form of text that describes in detail what conclusions you drew from each program-output pair.
Exercise 3: Skill 4.2. Different storage bindings are useful in different situations. This is why most language support more than one storage binding. For example, global variables in C use static, local variables (that are not explicitly marked “static”) use stack dynamic, and the variables allocated using “malloc” use explicit heap dynamic. For each of the following storage bindings, write a C or C++ program that demonstrates a realistic use of a variable that uses this storage binding. Discuss why your program exhibits a good use of the storage binding.

1. static
2. stack dynamic
3. explicit heap dynamic

Exercise 4: Skill 5.1. Consider the following Ada program.

```ada
procedure Main is
    X : Integer;
    procedure Sub1 is
        begin -- of Sub1
            Put(X);
        end; -- of Sub1
    procedure Sub2 is
        X : Integer;
        begin -- of Sub2
            X := 10;
            Sub1
        end; -- of Sub2
        begin -- of Main
            X := 5;
            Sub2
        end; -- of Main
```

1. Suppose the above program is compiled and executed using static scoping. What value of X is printed in procedure Sub1?

2. Under dynamic scoping, what value of X is printed in procedure Sub1?

(Sebesta, Chapter 5, Problem 7)

Exercise 5: Skill 5.2. Consider procedures A, B, C, D, and E. Let’s suppose that we have the following requirements:
A needs to call B and C
B needs to call C and E
C needs to call D
D needs to call E

Show how you would accomplish the above in a statically scoped language. You should try to organize things so that, to the best of your ability, all procedures that do not need to be called from a procedure must be hidden. Note that it may not be possible to do this completely: you just need to give it your best shot.

Exercise 6: Skill 5.3. The textbook defines two kinds of scoping: static and dynamic. Use

http://machine.cs.colorado.edu/~diwan/pldscoping.htm

to figure out if MYSTERY uses static or dynamic scoping (in this configuration). Your write up should describe the kind of scoping that MYSTERY uses and provide evidence for your conclusions. You may submit no more than three programs for this question. Programs that have parse errors (i.e., syntax errors) do not count. Every additional program will cost you 5% of the points for this question.

Exercise 7: Skill 5.3. Does Java use static or dynamic scoping? Support your answer either with quotes from the Java language definition or with examples and their output (much like how you use the PL-Detective) from a Java compiler.

Exercise 8: Synthesis. Some languages do all type checking at compile time; these are called statically-typed languages. Can a language that uses dynamic scoping be statically typed? Provide examples to support your answer as part of your explanation.