CS326 – Programming Languages, Concepts and Implementation

Homework 3

(Due February 27)

1. (20 pts) Consider the following fragment of code in C:

```
{     int a, b, c;
      ...
      {     int d, e;
            ...
            {     int f;
                  ...
            }
            ...
      }
      ...
      {     int g, h, i;
            ...
      }
      ...
}
```

   Assume that each integer variable occupies four bytes. How much total space is required for the variables in this code? Justify your answer.

2. (20 pts) Consider the following pseudocode, where procedures Q and R are nested inside procedure P:

```
procedure P (A,B: real)
      X: real

      procedure Q (B,C: real)
            Y: real
            ...

      procedure R (A,C: real)
            Z: real
            ...         // (*)
      ...
```

   Assuming static scope, what is the referencing environment (i.e., what names are known, and what do they refer to) at the location marked by (*)?

3. (30 pts) Consider the following pseudocode:

```
1. procedure main
2.    a: integer := 1
3.    b: integer := 2

4.    procedure middle
5.          b: integer := a

6.          procedure inner
7.                print a, b

8.          a: integer := 3

9.          // body of middle
10.         inner()
11.         print a, b

12.   // body of main
13.   middle()
14.   print a, b
```

(a) Suppose this was code for a language with the declaration-order rules of C (but with nested subroutines) - that is, names must be declared before use, and the scope of a name extends from its declaration through the end of the block. At each `print` statement, indicate which declarations of `a` and `b` are in the referencing environment. What does the program print (or will the compiler identify static semantic errors)?

(b) Repeat the exercise for the declaration-order rules of C# (names must be declared before use, but the scope of a name is the entire block in which it is declared).

(c) Repeat the exercise for the declaration-order rules Modula-3 (names can be declared in any order, and their scope is the entire block in which they are declared).

4. (30 pts) Consider the following pseudocode:

```
x: integer := 1
y: integer := 2

procedure add
      x := x + y

procedure second (P: procedure)
      x: integer := 2
      P()

procedure first
      y: integer := 3
      second(add)

// main program
```

```
first()
write integer(x)
```

(a) What does this program print if the language uses static scoping?
(b) What does it print if the language uses dynamic scoping with deep binding?
(c) What does it print if the language uses dynamic scoping with shallow binding?