CS 381, Spring 2024 Homework 5 (Scope and Parameter Passing)

Important Reminders!

1. Upload your solution as a text or pdf file to Canvas.
2. You must do all homework assignments by yourself, without the help of others. Also, you must not use services such as Chegg or Course Hero. If you need help, simply ask on Canvas, and we will help!
3. The homework is graded leniently, and we reward serious efforts, even when you can't get a correct solution.
4. You can work in teams of up to four students to create and submit homework assignments. Groups must be set up in advance and will have to be the same over the whole term. Each group submits one solution, and all group members receive the same grade for the homework.

Exercise 1. Runtime Stack

Consider the following block.

```
{ int x;
  int y;
  y := 1;
  { int f(int x) {
      if x=0 then {
         y := 1 }
      else  {
         y := f(x-1)*y+1 };
      return y
    };
    x := f(2);
  }
}
```

Illustrate the computations that take place during the evaluation of this block, that is, create a sequence of stacks, each showing the complete runtime stack with all activation records after each statement or function call.

Exercise 2. Static and Dynamic Scope

Consider the following block.

```
1   { int x;
2     int y;
3     int z;
4     x := 3;
5     y := 7;
6     { int f(int y) { return x*y };
7       int y;
8       y := 11;
9       { int g(int x) { return f(y) };
10        { int y;
11          y := 13;
12          z := g(2)
13        }
14      }
15    }
16  }
```

(a) Which value will be assigned to z in line 12 under static scoping?

(b) Which value will be assigned to z in line 12 under dynamic scoping?

It might be instructive to draw the runtime stack for different times of the execution, but it is not strictly required.

Exercise 3. Parameter Passing

What are the values of y and z at the end of the following block under the assumption that both parameters x are passed: (i) call by value, (ii) call by reference, and (iii) call by value-result.

```
{ int y;
  int z;
  y := 7;
  { int f(int x) {
      x := x+1;
      y := x;
      x := x+1;
      return y
    };
    int g(int x) {
      y := f(x)+1;
      x := f(y)+3;
      return x
    };
    z := g(y)
  }
}
```