HW5: Scope and Parameter Passing

David Kwak
kwakd@oregonstate.edu

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, draw a sequence of pictures each showing the complete runtime stack with all activation records after each statement or function call.

```
[x:?]
[y:?, x:?]
[y:1, x:?]
[f{}, y:1, x:?]
->
        [x:2, f{}, y:1, x:?]
        ->
                [x: 1, x:2, f{}, y:1, x:?]
                ->
                        [x:0 ,x: 1, x:2, f{}, y:1, x:?]
                        [ret:1, x:0 ,x: 1, x:2, f{}, y:1, x:?]
                <-
                [x:1, x:2, f{}, y:2, x:?]
                [ret:2, x:1, x:2, f{}, y:5, x:?]
        <-
        [x:2, f{}, y:5, x:?]
        [ret:5, x:2, f{}, y:5, x:?]
<-
[f:{}, y:5, x:5]
[y:5, x:5]
[]
```

Exercise 2. Static and Dynamic Scope

Consider the following block.

```
{   int x;
    int y;
    int z;
    x := 3;
    y := 7;
   { int f(int y) { return x*y};
     int y
     y := 11;
     { int g(int x) { return f(y)};
        { int y;
          y := 13;
          z := g(2)
        }
     }
   }
}
```

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

[x:?]

[y:?, x:?]

[z:?, y:?, x:?]

[z:?, y:7, x:3]

[f{}, z:?, y:7, x:3]

[y:?, f{}, z:?, y:7, x:3]

[y:11, f{}, z:?, y:?, x:3]

[g{}, y:11, f{}, z:?, y:7, x:3]

[y: ?, g{}, y:11, f{}, z:?, y:7, x:3]

[y: 13, g{}, y:11, f{}, z:?, y:7, x:3]

[z:?, y: 13, g{}, y:11, f{}, z:?, y:7, x:3]

[x:2 , z:?, y: 13, g{}, y:11, f{}, z:?, y:7, x:3]

[**y:11**, x:2 , z:?, y: 13, g{}, y:11, f{}, z:?, y:7, **x:3**]

z := 33 (y:11 * x:3)

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

[x:?]

[y:?, x:?]

[z:?, y:?, x:?]

[z:?, y:7, x:3]

[f{}, z:?, y:7, x:3]

[y:?, f{}, z:?, y:7, x:3]

[y:11, f{}, z:?, y:?, x:3]

[g{}, y:11, f{}, z:?, y:7, x:3]

[y: 13, g{}, y:11, f{}, z:?, y:7, x:3]

[z:?, y: 13, g{}, y:11, f{}, z:?, y:7, x:3]

[x:2 , z:?, y: 13, g{}, y:11, f{}, z:?, y:7, x:3]

[**y:13**, **x:2** , z:?, y: 13, g{}, y:11, f{}, z:?, y:7, x:3]

z := 26 (y:13 * x:2)

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);
   }
}
```

(i) call by value

```
[z:?, y:?]
[z:?, y:7]
[f{}, z:?, y:7]
[g{}, f{}, z:?, y:7]
-> g()
        [x:7 ,g{}, f{}, z:?, y:7]
        -> f()
                [x:7 , x:7 ,g{}, f{}, z:?, y:7]
                [x:8 , x:7 ,g{}, f{}, z:?, y:7]
                [x:8 , x:7 ,g{}, f{}, z:?, y:8]
                [ret: 8, x:8 , x:7 ,g{}, f{}, z:?, y:8]
        <-
        [x:7 ,g{}, f{}, z:?, y:8]
        [x:7 ,g{}, f{}, z:?, y:9]
        ->f()
                [x:9 , x:7 ,g{}, f{}, z:?, y:9]
                [x:10 , x:7 ,g{}, f{}, z:?, y:9]
                [x:10 , x:7 ,g{}, f{}, z:?, y:10]
                [x:11 , x:7 ,g{}, f{}, z:?, y:10]
                [ret: 10, x:11 , x:7 ,g{}, f{}, z:?, y:10]
        <-
        [x:10 ,g{}, f{}, z:?, y:10]
```

```
    [x:13 ,g{}, f{}, z:?, y:10]
    [ret: 13, x:13 ,g{}, f{}, z:?, y:10]
<-
[g{}, f{}, z:13, y:10]
y:= 10
z:= 13
```

(ii) call by reference

```
[z:?, y:?]
[z:?, y:7]
[f{}, z:?, y:7]
[g{}, f{}, z:?, y:7]
-> g()
    [x->y ,g{}, f{}, z:?, y:7]
    -> f()
        [x->x, x->y ,g{}, f{}, z:?, y:7]
        [x->x, x->y ,g{}, f{}, z:?, y:8]
        [x->x, x->y ,g{}, f{}, z:?, y:8]
        [ret: 9, x->x, x->y ,g{}, f{}, z:?, y:9]
    <-
    [x->y, g{}, f{}, z:?, y:9]
    [x->y, g{}, f{}, z:?, y:10]
    ->f()
        [x->x, x->y ,g{}, f{}, z:?, y:10]
        [x->x, x->y ,g{}, f{}, z:?, y:11]
        [x->x, x->y ,g{}, f{}, z:?, y:11]
        [x->x, x->y ,g{}, f{}, z:?, y:12]
        [ret: 12, x->x, x->y ,g{}, f{}, z:?, y:12]
    <-
    [x->y, g{}, f{}, z:?, y:12]
    [x->y, g{}, f{}, z:?, y:15]
    [ret: 15, x:y ,g{}, f{}, z:?, y:15]
<-
[g{}, f{}, z:15, y:15]
y:= 15
z:= 15
```

(iii) call by value-result

```
[z:?, y:?]
[z:?, y:7]
[f{}, z:?, y:7]
[g{}, f{}, z:?, y:7]
-> g()
```

```
    [x:7 ,g{}, f{}, z:?, y:7]
    -> f()
        [x:7 , x:7 ,g{}, f{}, z:?, y:7]
        [x:8 , x:7 ,g{}, f{}, z:?, y:7]
        [x:8 , x:7 ,g{}, f{}, z:?, y:8]
        [x:9 , x:7 ,g{}, f{}, z:?, y:8]
        [ret: 8, x:9 , x:7 ,g{}, f{}, z:?, y:8]
        [ret: 8, x:9 , x:8 ,g{}, f{}, z:?, y:8]
    <-
    [x:8 ,g{}, f{}, z:?, y:8]
    [x:8 ,g{}, f{}, z:?, y:9]
    ->f()
        [x:9 , x:8 ,g{}, f{}, z:?, y:9]
        [x:10 , x:8 ,g{}, f{}, z:?, y:9]
        [x:10 , x:8 ,g{}, f{}, z:?, y:10]
        [x:11 , x:8 ,g{}, f{}, z:?, y:10]
        [ret: 10, x:11 , x:8 ,g{}, f{}, z:?, y:10]
    <-
    [x:8 ,g{}, f{}, z:?, y:10]
    [x:13 ,g{}, f{}, z:?, y:10]
    [ret: 13, x:13 ,g{}, f{}, z:?, y:10]
    [ret: 13, x:13 ,g{}, f{}, z:?, y:13]
<-
[g{}, f{}, z:13, y:13]
y:= 13
z:= 13
```