프로그래밍언어 HW#2 20191629 이주희

1. Some programming languages are typeless. What are the obvious advantages and disadvantages of having no types in a language?

Most of advantages of typeless languages rise from their flexibility of data manipulation. However this flexibility is also a disadvantage in some area, as the compiler or interpreter cannot verify the integrity of data before executing the code.

2. Consider the following JavaScript program. List all variables.

```
var x, y, z;
function sub1() {
    var a, y, z;
    function sub2() {
        var a, b, z;
    }
}
function sub3() {
    var a, x, w;
}
```

Note that JavaScript employs "variable hoisting", where all variables with var keyword are hoisted to the top of their enclosing function.

Global scope: x_1, y_1, z_1

In sub1: a_1, x_1(global), y_2(local), z_2(local)

In sub2: a_2(local), b, x_1(global), y_2(sub1), z_2(sub1)

In sub3: a_3(local), x_2(local), y_1(global), z_1(global), w.

3. Consider the following Python program. List all variables.

```
x=1; y=3; z=5
def sub1():
    a=7; y=9; z=11
def sub2():
    global x
    a=13; x=15; w=17
    def sub3():
        nonlocal a
        a=19; b=21; z=23
```

Python declares new variables if they are not already present in their local scope.

Global scope: x_1, y_1, z_1

In sub1: a_1(local), x_1(global), y_2(local), z_2(local)

In sub2: a_2(local), x_1(global), y_1, z_1, w

In sub3: a_2(sub2), b, x_1(global), y_1, z_2, w

4. Consider a skeletal C program with dynamic scoping. What variables are visible?

a. main → fun1 → fun2 → fun3.
   a(main), b(fun1), c(fun2), d(fun3), e(fun3), f(fun3).

b. main → fun1 → fun3.
   a(main), b(fun1), c(fun1), d(fun3), e(fun3), f(fun3)

c. main → fun2 → fun3 → fun1
   a(main), b(fun1), c(fun1), d(fun1), e(fun3), f(fun3)

d. main → fun3 → fun1
   a(main), b(fun1), c(fun1), d(fun1), e(fun3), f(fun3)

프로그래밍언어 HW#2 (cont'd)

e. main → fun1 → fun3 → fun2

a(main), b(fun1), c(fun2), d(fun2), e(fun2), f(fun3)

f. main → fun3 → fun2 → fun1

a(main), b(fun1), c(fun1), d(fun1), e(fun2), f(fun3)

5. Consider a program written in a JavaScript-like syntax. What variables are visible?

a. main → sub1 → sub2 → sub3

a(sub3), b(sub2), x(sub3), y(sub1), z(sub2), w(sub3)

b. main → sub1 → sub3

a(sub3), x(sub3), y(sub1), z(sub1), w(sub3)

c. main → sub2 → sub3 → sub1

a(sub1), b(sub2), x(sub3), y(sub1), z(sub1), w(sub3)

d. main → sub3 → sub1

a(sub1), x(sub3), y(sub1), z(sub1), w(sub3)

e. main → sub1 → sub3 → sub2

a(sub2), b(sub2), x(sub3), y(sub1), z(sub2), w(sub3)

f. main → sub3 → sub2 → sub1

a(sub1), b(sub2), x(sub3), y(sub1), z(sub1), w(sub3)