<Homework 2> Solve the Problems Chapter 5

20201635 컴퓨터공학과 전찬

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

 typeless한 언어의 가장 큰 장점은 속도가 빠르다는 것이다. type check 등의 compile에서 타입에 관련해서 필요한 과정들이 모두 생략되기 때문에, type이 존재하는 언어들보다 훨씬 빠른 속도로 프로그램을 만들어낼 수 있다. 하지만 typeless이기 때문에 자동으로 그 언어에서 type check를 수행해주지 않으며, 프로그램을 작성하는 사람에게 더 많은 책임을 부여한다. 따라서 프로그래밍 언어의 reliability가 낮으며, 프로그램을 작성하는 사람이 더 신중할 필요가 있다는 단점이 존재한다.

#2. Consider the following JavaScript program: List all the variables, along with the program units where they are declared, that are visible in the bodies of sub1, sub2, and sub3, assuming static scoping is used.

sub1()에서 a, x, y, z을 visible할 수 있으며, 이 중 a, y, z는 sub1()에서 정의된 변수, x는 global에서 정의된 변수이다. sub2()에서는 a, b, x, y, z을 visible 할 수 있으며, a, b, z는 sub2() 내부에서 정의된 변수, y는 sub1()에서 정의된 함수, x는 global로 정의된 변수이다. 마지막으로 sub3()에서는 a, x, y, z, w을 visible할 수 있으며, a, x, w는 sub3() 내부에서 정의된 변수, y, z는 global로 정의된 변수이다.

#3. Consider the following Python program. List all the variables, along with the program units where they are declared, that are visible in the bodies of sub1, sub2, and sub3, assuming static scoping is used.

sub1()에서 a, x, y, z 를 visible할 수 있으며, a, y, z는 sub1()에서 정의된 변수, x는 global로 정의된 변수이다. sub2()에서는 a, x, y, z, w를 visible할 수 있으며, a, w는 sub2()에서 정의된 변수, x, y, z는 global로 정의된 변수이다. 마지막으로 sub3()에서는 a, b, x, y, z, w를 visible할 수 있으며, b, z는 sub3()에서 정의된 변수, a, w는 sub2()에서 정의된 변수, x, y는 global로 정의된 변수이다.

추가로 global keyword는 현재 global variable로 존재하는 변수를 연결하게 되며, nonlocal은 global이 아닌 변수들 중, nonlocal keyword가 사용된 scope의 바로 바깥 scope에 해당하는 변수를 연결할 수 있는 keyword 이다.

#4. Consider the following skeletal C program. Given the following calling sequences and assuming that dynamic scoping is used, what variables are visible during execution of the last function called? Include with each visible variable the name of the function in which it was defined.

a. main calls fun1; fun1 calls fun2; fun2 calls fun3.

a in main function, b defined in fun1(), c in func2(), d, e, f in func3()

b. main calls fun1; fun1 calls fun3.

a in main function, b, c in func1(), d, e, f in func3().

c. main calls fun2; fun2 calls fun3; fun3 calls fun1.

a in main function, b, c, d in func1(), e, f in func3().

d. main calls fun3; fun3 calls fun1.

a in main function, b, c, d in func1(), e, f in func3().

e. main calls fun1; fun1 calls fun3; fun3 calls fun2.

a in main function, b in func1(). c, d, e in func2(), f in func3().

f. main calls fun3; fun3 calls fun2; fun2 calls fun1.

a in main function, b, c, d in func1(). e in func2(). f in func3().

#5. Consider the following program, written in JavaScript-like syntax. Given the following calling sequences and assuming that dynamic scoping is used, what variables are visible during execution of the last subprogram activated? Include with each visible variable the name of the unit where it is declared.

a. main calls sub1; sub1 calls sub2; sub2 calls sub3.

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

b. main calls sub1; sub1 calls sub3.

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

c. main calls sub2; sub2 calls sub3; sub3 calls sub1.

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

d. main calls sub3; sub3 calls sub1.

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

e. main calls sub1; sub1 calls sub3; sub3 calls sub2.

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

f. main calls sub3; sub3 calls sub2; sub2 calls sub1.

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