**Test #8. 2012.4.30 프로그래밍언어론 00 분반- 개인 문제**

학번:

이름:

조번호:

1. 다음 각 항목이 참이면 O, 아니면 X 를 하시오.
2. Python 에서는 함수호출을 할 때 actual parameter 의 순서가 함수의 formal parameter 와 동일해야 한다.
3. Default parameter는 actual parameter 가 전달되지 않더라도 함수 호출이 수행가능 하도록 해준다.
4. Python 에서 def foo(p1, p2, \*a, \*\*h); . 로 정의된 함수에 대해 foo(2, 4, 6, 8, mon=68, tue=72, wed=77) 로 호출하면, a에 해당하는 것은 [6,8]이 된다.
5. 함수는 일반적인 프로시져와 달리, 호출시 늘 실행 결과값 하나를 리턴하게 된다.
6. static 지역 변수가 없는 함수는, 동일한 parameter를 가지고 여러 번 호출하는 경우 항상 동일한 값을 리턴하는 것을 보장할 수 없다.
7. C에서 n개의 int 필드를 가지는 struct A를 리턴타입으로 하는 함수를 호출하면, 리턴 값 전달 시간은 n이 커질수록 커지게 된다.
8. call-by-value-result 또는 pass-by-value-result 는 inout model 의 parameter 전달에 사용된다.
9. call-by-name은 actual parameter 이름 자체가 전달되어 formal parameter로 대치되며, readability 향상을 위해 사용한다.
10. Python에서는 scalar 변수가 parameter로 전달 될떄는 call-by-value가, 배열 등이 전달될 때는 call-by-reference를 사용한다.
11. 다음 int foo(int f[][]) {… }과 같은 함수선언은 C, C++에서는 가능하나 Java에서는 가능하지 않다.

2. pass-by result 에서 formal parameter가 actual parameter 에 값이 전달되는 순서에 따라 결과가 달라지는 코드 예를 보이고 간단히 설명하시오.

|  |
| --- |
| procedure SUB1;  var x: integer;  procedure SUB2;  begin  write('x =', x);  end;  procedure SUB3(subx);  var x: integer;  begin  x := 3;  call subx;  end;  begin  x := 1;  call sub3(SUB2);  end; |

3. 오른쪽 프로그램에서 parameter subprogram의 실행환경을 shallow binding, deep binding 으로 지정할 때 출력을 각각 적으시오.

shallow binding :

deep binding :

**Test #8. 2012.4.30 프로그래밍언어론 00 분반 조별 문제**

조번호:

참여자 (학번, 이름)

-

-

-

-

-

1. 연상

2.다음 프로그램은 함수를 리턴하는 함수에 대한 것이다. 리턴되는 함수는 static scoping rule을 적용받는다

def counter():

n = 0

def increment(x):

nonlocal n

n += x

print(n)

return increment

이 함수는 다음과 같이 호출하면, 1,7,1,8을 리턴한다는 것을 Test#4 조별문제에서 파악하였다.

c1 = counter()

c2 = counter()

c1(1)

c1(5)

c2(1)

c1(1)

위 프로그램을 수정한 다음과 같은 코드에서

def counter():

n = 0

def increment(x):

nonlocal n

n += x

**x++;**

print(n)

return increment

i = 0

c1 = counter()

c2 = counter()

c1(**i**)

c2(**i**)

c1(**i**)

의 출력값을 (1) call-by-value일 때, (2) call-by-reference, (3) call-by-name 일 때 각각 구하시오.