

Programming Language Assignment #2

External Document

소프트웨어대학 소프트웨어학부

20204043 문버리



#코드실행방법

1. 문서 안에 들어있는 venv 안의 main.py 함수를 실행합니다.
2. 1 번 줄의 open 함수에서

```
code = open('./../code1.txt')
```

경로를 바꾸어줍니다.

Code1 부분을 code2, code3, code4.....로 바꾸면 다른 텍스트파일이 실행이 됩니다.

#과제 예제 코드

code1.txt 파일에 있습니다.

코드:

first {

variable a, b, c;

call second;

p;

}

second {

variable x, a;

b;

print_ari;

q;

}

main {

variable p, q;

```
    call first;
}
```

컴파일 결과

```
/usr/local/bin/python3.7 /Users/PC/Downloads/과제/venv/main.py
Syntax 0.K.

second : b => 1 , 3

second :
Local variable: a
Local variable: x
Dynamic Link: 2
Return Address: {'first': 1}
first :
Local variable: c
Local variable: b
Local variable: a
Dynamic Link: 0
Return Address: {'main': 1}
main :
Local variable: q
Local variable: p

second : q => 2 , 1

first : p => 1 , 0

Process finished with exit code 0
```

#추가 코드 2

-정의되지 않은 함수를 호출한 경우 에러처리

code2.txt 파일에 있습니다.

코드:

```
first {  
  
    variable a, b, c;  
  
    call second;  
  
    p;  
  
}  
  
second {  
  
    variable x, a;  
  
    b;  
  
    print_ari;  
  
    q;  
  
}  
  
main {  
  
    variable p, q;  
  
    call first;  
  
    call third;  
  
}
```

컴파일 결과:

```
Syntax 0.K.
```

```
second : b => 1 , 3
```

```
second :
```

```
Local variable: a
```

```
Local variable: x
```

```
Dynamic Link: 2
```

```
Return Address: {'first': 1}
```

```
first :
```

```
Local variable: c
```

```
Local variable: b
```

```
Local variable: a
```

```
Dynamic Link: 0
```

```
Return Address: {'main': 1}
```

```
main :
```

```
Local variable: q
```

```
Local variable: p
```

```
second : q => 2 , 1
```

```
first : p => 1 , 0
```

```
Call to undefined function: third
```

```
Process finished with exit code 0
```

#추가 코드 3

-main 함수 없는경우 에러처리

code3.txt 파일에 있습니다.

코드:

```
first {  
  
    variable a, b, c;  
  
    call second;  
  
    p;  
  
}  
  
second {  
  
    variable x, a;  
  
    b;  
  
    print_ari;  
  
    q;  
  
}  
  
third {  
  
    variable p, q;  
  
    call first;  
  
}
```

컴파일 결과:

```
/usr/local/bin/python3.7 /Users/PC/Downloads/과제/venv/main.py
```

```
Syntax 0.K.
```

```
No starting function.
```

```
Process finished with exit code 0
```

#추가 코드 4

-변수와 함수가 같은 이름인 경우 에러처리

code4.txt 파일에 있습니다.

코드:

```
first {  
  
    variable a, b, c;  
  
    call second;  
  
    p;  
  
}  
  
second {  
  
    variable x, a;  
  
    b;  
  
    print_ari;  
  
    q;  
  
}  
  
main {  
  
    variable p, q, first;  
  
    call first;  
  
}
```

컴파일 결과:


```
/usr/local/bin/python3.7 /Users/PC/Downloads/과제/venv/main.py  
Syntax 0.K.  
Duplicate declaration of the identifier or thr function name: first  
  
Process finished with exit code 0
```

처리한 에러: 중복된 연산자를 제거하고 파싱을 완료했다.

#추가 코드 5

-함수명 중복선언 에러처리

code5.txt 파일에 있습니다.

코드:

```
first {  
  
    variable a, b, c;  
  
    p;  
  
}  
  
first {  
  
    variable x, a;  
  
    b;  
  
    print_ari;  
  
    q;  
  
}  
  
main {  
  
    variable p, q;  
  
    call first;  
  
}
```

컴파일 결과:

```
Syntax O.K.  
Duplicate declaration of the function name:  first  
  
Process finished with exit code 0
```

#추가 코드 6

-변수 중복 선언 에러처리

code6.txt 파일에 있습니다.

코드:

```
first {  
  
    variable a, b, c;  
  
    call second;  
  
    p;  
  
}  
  
second {  
  
    variable x, a;  
  
    b;  
  
    print_ari;  
  
    q;  
  
}  
  
main {  
  
    variable p, q;  
  
    variable p;  
  
    call first;  
  
}
```

컴파일 결과:

```
Syntax O.K.  
Duplicate declaration of the identifier:  p  
  
second : b => 1 , 3  
  
second :  
Local variable:  a  
Local variable:  x  
Dynamic Link:  2  
Return Address: {'first': 1}  
first :  
Local variable:  c  
Local variable:  b  
Local variable:  a  
Dynamic Link:  0  
Return Address: {'main': 2}  
main :  
Local variable:  q  
Local variable:  p  
  
second : q => 2 , 1  
  
first : p => 1 , 0  
  
Process finished with exit code 0
```

#추가 코드 7

-first 함수에서도 print_ari 호출할 경우

code7.txt 파일에 있습니다.

코드:

```
first {  
  
    variable a, b, c;  
  
    call second;  
  
    print_ari;  
  
}
```

```
second {  
  
    variable x, a;  
  
    b;  
  
    print_ari;  
  
    q;  
  
}
```

```
main {  
  
    variable p, q;  
  
    call first;  
  
}
```

컴파일 결과:

Syntax O.K.

second : b => 1 , 3

second :

Local variable: a

Local variable: x

Dynamic Link: 2

Return Address: {'first': 1}

first :

Local variable: c

Local variable: b

Local variable: a

Dynamic Link: 0

Return Address: {'main': 1}

main :

Local variable: q

Local variable: p

second : q => 2 , 1

first :

Local variable: c

Local variable: b

Local variable: a

Dynamic Link: 0

Return Address: {'main': 1}

main :

Local variable: q

Local variable: p

#추가 코드 8

-함수 내에 있는 지역 변수를 호출한 경우

code8.txt 파일에 있습니다.

코드:

```
first {  
  
    variable a, b, c;  
  
    a;  
  
}  
  
main {  
  
    variable p, q;  
  
    call first;  
  
}
```

컴파일 결과:

```
Syntax O.K.
```

```
first : a => 0 , 2
```

#추가 코드 9

-더 많은 함수로 늘어난 경우

code9.txt 파일에 있습니다.

코드:

```
last{  
    d;  
    print_ari;  
}  
third {  
    call last;  
    d;  
}  
second {  
    variable b, c, d;  
    call third;  
}  
first {  
    variable a, b, c;  
    call second;  
}  
main {  
    variable p, q;  
    call first;  
}
```


컴파일 결과:

```
Syntax O.K.

last : d => 2 , 4

last :
Dynamic Link: 12
Return Address: {'third': 0}
third :
Dynamic Link: 7
Return Address: {'second': 1}
second :
Local variable: d
Local variable: c
Local variable: b
Dynamic Link: 2
Return Address: {'first': 1}
first :
Local variable: c
Local variable: b
Local variable: a
Dynamic Link: 0
Return Address: {'main': 1}
main :
Local variable: q
Local variable: p

third : d => 1 , 4
```

#추가 코드 10

-세미콜론 빠뜨린 경우 실택스에러처리하고 종료

code10.txt 파일에 있습니다.

코드:

```
first {  
    variable a, b, c  
}  
  
main {  
    variable p, q;  
    call first;  
}
```

컴파일 결과:

Syntax error

#추가 코드 11

-닫는 괄호 빼먹었을 경우 실택스에러처리하고 종료

code11.txt 파일에 있습니다.

코드:

```
first {  
  
    variable a, b, c;  
  
}  
  
main {  
  
    variable p, q;  
  
    call first;
```

컴파일 결과:

```
Syntax error
```

```
Process finished with exit code 0
```

#추가 코드 12

-변수 선언 빠뜨린 경우 실행스케치 처리하고 종료

code12.txt 파일에 있습니다.

코드:

```
first {  
  
    variable a, , c;  
  
}  
  
main {  
  
    variable p, q;  
  
    call first;  
  
}
```

컴파일 결과:

```
Syntax error
```

```
Process finished with exit code 0
```