12.11. Flow of Execution Summary

When you are working with functions it is really important to know the order in which statements are executed. This is called the **flow of execution** and we've already talked about it a number of times in this chapter.

Execution always begins at the first statement of the program. Statements are executed one at a time, in order, from top to bottom. Function definitions do not alter the flow of execution of the program, but remember that statements inside the function are not executed until the function is called. Function calls are like a detour in the flow of execution. Instead of going to the next statement, the flow jumps to the first line of the called function, executes all the statements there, and then comes back to pick up where it left off.

That sounds simple enough, until you remember that one function can call another. While in the middle of one function, the program might have to execute the statements in another function. But while executing that new function, the program might have to execute yet another function!

Fortunately, the Python interperter is adept at keeping track of where it is, so each time a function completes, the program picks up where it left off in the function that called it. When it gets to the end of the program, it terminates.

What does all that mean for us when we try to understand a program? Don't read from top to bottom. Instead, follow the flow of execution. This means that you will read the def statements as you are scanning from top to bottom, but you should skip the body of the function until you reach a point where that function is called.

Check your understanding

func-10-1: Consider the following Python code. Note that line numbers are included on the left.

```
1  def pow(b, p):
2      y = b ** p
3      return y
4
5  def square(x):
6      a = pow(x, 2)
7      return a
8
9  n = 5
10 result = square(n)
11 print(result)
```

What does this function print?

- A. 25
- B. 5
- C. 125
- D. 32

Check me | Compare me

✔ The function square returns the square of its input (via a call to pow).

Activity: 1 -- Multiple Choice (question11_10_1)

12.10. Functions can call other functions (Composition)">
unctions can call other functions (Composition)">
12.12. Print vs. return">
You have attempted 2 of 1 activities on this page

| Back to top