Title: Notes, Week 6
Page 1 of 1

Name: Zach Caldwell
Class: CS 112-01
Date: Feb. 14, 2017

Cues:	**Notes:**
	1. Functions:
Functions description	a. Are like a “mini-program within a program,” i.e. a subset of instructions that run when specifically called.
Parameters & arguments	b. Can optionally name one or more ‘parameter’ variables in their function signature that can be used to manipulate the function’s execution; anything passed in to a parameter from the caller is known as an ‘argument’.
Keyword arguments	c. Can optionally take ‘keyword’ arguments that allow one to name certain parameters when passing arguments.
Syntax description	d. Are defined using the ‘def’ statement, a function name, a list of parameters inside parentheses, a colon, and a code block.
Return keyword	e. Can send data back to the calling statement using the ‘return’ keyword and a value or expression.
Default value / NoneType	f. Return the special ‘None’ value of the NoneType data type by default if no other return value is specified.

Syntax example	g. Syntax example:
	`>>> def plusone(num):`
	`>>>     return num + 1`

Variable scope	h. “Parameters and variables that are assigned in a called function are said to exist in that function’s *local scope*,” and are destroyed / inaccessible whenever program flow is outside the function. Variables outside of any function or class exist in the *global scope* and are universally accessible within the program until it exits.
Variable name scopes	i. Variable names can be the same between global and local scope, or between multiple local scopes, but it should be avoided because they are ultimately different variables.

Comments: I wasn’t sure whether we were covering things like exception handling and imports this week, so I’ve just noted on functions here.