Week Three Reading Notes

This week may be the most critical for mastering Python programming. After this week, you should be able to write some interesting programs.

Lecture Sequence 5

The first video segment discusses loops and should be straightforward to follow. The next segment introduces recursion, which is a fancy word, but the idea is very natural. Watching the lectures on loops and recursion, I thought about the song we used to sing when we were young and went on long school trips: "99 bottles of beer on the wall. If one of them happens to fall, there will be 98 bottles of beer on the wall." The song then continues anew. I do not think singing the next round of the song is any simpler than the first round, but it is certainly simpler to hold ten bottles of beer than 99. The lecture uses the term "simpler" in the same way. At first, I did not understand what was meant by something being simpler, but the song helped me understand.

If you find the segment on recursion a little bit confusing, don't worry. The next segment will explain things in more detail. It goes through the code step by step. By now, the pattern should be clear. The theory comes first and it may seem too abstract. The code walkthough that follows, however, should clear up any confusion.

When you understand the example of recursive multiplication, it may be helpful to stop for a moment and think about it. Do you understand why we say that recursion is a different way of thinking than iteration?

The lectures get into the details of mathematical induction. This is typically taught in high school. If you know it, great - consider this a chance to review. If not, that is also fine because the lecture is clear and takes the mystery out of the fancy terms of "induction," "proof," and "algebra." Remember, this is a computer science course and not just programming. We know you can understand this, so take it step by step.

Once you can understand how to prove that a loop or recursion terminates, your programs will have fewer bugs.

Prof Grimson talks about nice and crisp code. What words would you use to describe it? It is very hard to build an automatic grader that can identify crisp, clean, nice code. It should already be apparent that there are many different programs that solve the same problem, but not all of them are crisp or beautiful.

After the finger exercises, things get interesting. The Tower of Hanoi example should make recursion much easier to understand. The Fibonacci example will help you appreciate the usefulness of recursion. Be sure to follow the Fibonacci example, even if you have seen it before, because the Python "assert" command is slipped into the example code.

Please, if you are starting to get weary of all the mathematical examples, do not give up. You now know enough Python that some interesting non-mathematical examples can be shown.

Lecture Sequence 6

We are almost finished with the somewhat tedious foundations. After Lecture 6, we will have enough tools to write interesting programs. You will be surprised by how much you have learned already!

A word of caution about some of the words used in the lectures:

* "Techs": The example uses the variable "Techs" as shorthand for Institutes of Technology. Listen carefully as it sounds a bit too much like the word "Text."

* "Mutable": The word "mutable" may not be familiar to non-native English speakers. It is similar to the word "mutation" and means that the values **can** be changed. Also note that the word "immutable" means the opposite – values **cannot** be changed.

* "Tuples": Just a set of things, similar to, but not the same thing as "lists" -- both are fundamental objects in Python and it may take a bit of reflection to understand the differences. The major difference is that tuples, unlike lists, are immutable.

* "Lists": Another important Python object. The file system on you computer is something like a list. You can have files or folders. Inside a folder can be files or other folds or both.

* "Aliases": Two distinct paths to the same object. It is the same thing as used in file system, and sometimes called a "shortcut".

* "Range": It is a very important Python function and is defined in L6 Problem 4. You first saw this function in L3 Problem 4.

* "Method": When talking about lists, methods are introduced. Pay attention as they are very important.

* "Cloning": Cloning of lists is a concept that is also worth understanding.

A note about Problem Set 3: In this problem set, you will write a program to play the game Hangman. Writing this program is really fun but it will take some time and effort. It will help you learn how to go about writing a larger program, and also give you something really cool to show off to friends and family! Be sure to give yourself plenty of time to tackle Problem Set 3.

-Larry Rudolph