Week 9, Lecture 1

In this lecture we look at examples written using the author's graphics.py module. Copy and save this module and place it in the same folder that contains your graphics function calls. The examples are based on the material in Chapter 4. Because these examples were written when the chapters were being followed in order, the examples make minimal or no use of function calls and lists etc. So one ready exercise for you to do is to try and rewrite some or all of the examples using function calls. In doing that, you will discover at least two things: first, the code will become much, much cleaner, and second, you will be able to package the construction of graphical objects in your own creative ways. Apart from these two things, you will quickly become comfortable with Python graphics. Graphics work tends to be detailed, and the author recognized that providing you with a subset of Tkinter (Python Graphics) in the form of graphics.py would give you an easier way to get into both Python programming and graphics.

You may not realize it at first, but each time you construct a graphical object you are doing Object-Oriented Programming. You were already doing it with numbers, strings and lists, but now it becomes more readily apparent because circles, squares and triangles are objects that you can see and move around within a window on the screen. So, to begin, we will look at how to construct a graphical window, and draw various objects in this window, along with grid lines, some simple-minded animation (to make you think about how animation really works) and how to draw a bar chart without asking Matplotlib for help. Once you see how to do this you'll begin to see what the code inside Matplotlib must look like.

Graphics can be fun, though time-consuming. Patience is a requirement, but then it is always a requirement in learning. Avoid thinking in terms of short-cuts when you learn. Always try to enjoy the process. For your homework, play around with various graphical objects and try to package the lecture code in neat functions that you can reuse elsewhere. For example, how can you write a function that helps you draw a bar graph with n bars? You'll need some parameters in the function call: bar colors, heights, thicknesses etc.