**Lab 2:**

SecondConverter – I found this pretty easy to implement. It is also useful to show off Python’s capability to substitute %d for an int in a string you want to print. For example: print "%d seconds is equal to %d hours, %d minutes, and %d seconds." %(totSec,hours,minutes,sec) is used to print the output of the converter, which is a bit more intuitive than Java’s concatenation, I think.

Beer – This one was kind of weird. I could be wrong, and as I go on I’m sure I will find something, but my impression is that Python doesn’t support class constants. It is fairly easy however, to just use the same syntax for class constants to define a variable and treat it as a class constant. I imagine that when we get to the more complicated programs we will miss Java’s functionality in this regard. Python has an interesting for loop structure, you can use range(a,b), range(b), or range(a, b, c) which correspond to a list from a to b, a list from 0 to b, and a list from a to b, being incremented by c. This may save time, when you so often just need an increment of one, you don’t have to explicitly specify that in the for loop.

Sequence – Again, I just used a variable as my ‘class constant.’ Other than that, this was straightforward to implement. One thing is that to make Python do the equivalent of a print, rather than a println function, you have to add a comma after your whole print statement. For example: print “example”, is the same as System.out.print(“example”), while print “example” is the same as System.out.println(“example”). This is somewhat confusing.

Interest – This one was basically straightforward as well. I spent time trying to figure out why my floats weren’t being displayed as floats before I realized that I had the wrong placeholder in the string. A nice touch for Python is that you can specify how many decimals you want to use in your float. For example, putting %.2f in your string will display the float to 2 decimal places.