PHYS 5061 Lab 2: More getting up to speed with LabVIEW

(1) Work through chapter 3 of Essick, mastering the FOR loop and the waveform graph.

(2) Complete and save the VI for the final do-it-yourself coin flipper VI at the end of chapter 3. You may skip the "Use It!" It's worth reading though.

(3) Modify it (the DIY VI) to do 50 coins in a "toss" and save as a new VI. (Nevermind creating 50 heads/tails indicators for this situation! That'd be cruel. Simulating a flip of 50 coins can be done with a loop within a loop – doing each one of the 50 coins at a time, rather than flipping all 50 "in parallel.") To total up the number of heads, there is a useful hint in problem 1 of chapter 3 on summing elements in an array, or you can look ahead to chapter 8, where *shift registers* are introduced as a means of retaining and updating a value from one loop iteration to the next. You only need to look at the first two pages of chapter 8 to get this. There's also an advanced/general histogram VI that you may want to explore as a somewhat nicer (more controlled) way to get the histogram.

Comment this VI by adding text to the block diagram explaining to someone what/how this VI does its thing. And add a brief instruction to the front panel on how to use it for someone who picks it up to try six weeks from now.

(4) (a) Complete problem #3 from chapter 3 and save the VI. Same requirement on commenting/documenting your VI. (b) Create a VI that extends the series sum in (a) to contain a user-specified number of terms and updates the plot after each term is added. You probably want to put a delay in somewhere so you can see successive plots at a human-friendly pace. And you may really, really want to look at that shift register mumbo jumbo mentioned above now.

Print out the VI documentation for your various VI's and put them in your lab notebook with any additional comments you may have.

Remember: After you have finished a VI and *think* it's perfect, make a final test of it by closing it, loading it afresh into LabVIEW and see if it performs as desired. You may discover LabVIEW has decided on assigning some default conditions that make for unexpected behavior.