Project 6 Addendum:

This addendum to the N2T book project description takes precedence where there is any duplication. The changes correct some design choices which just aren't very clean and removes the necessity to unnecessarily twiddle with other types state data. The changes are also more "Pythonic" and follow good naming conventions making it easier to read and use the provided code in context of what you have learned so far.

Start with the provided code, it includes working unit test cases. Your modifications must pass the provided unit tests to have a chance to work properly when connected to each other. I will demonstrate how to run and interpret unit tests in class. These unit tests are not guaranteed exhaustive (it may be possible to find a way to write code that passes unit tests but still has an unchecked problem, over time we strive to eliminate those cracks.)

Replace the contents of your **`projects/yourLastName06`** directory with the contents of the start code zip, it is all inclusive.

The included **`.py`** files ares slightly different than the plan presented in the book, as previously mentioned these file's organization takes precedence. **`SymbolTable.py`** and **`Code.py`** are complete as provided and require no changes. We will discuss them and why these particular design choices are so useful in class.

Comment out your tracking/debugging comments. Not with ## a the beginning of the line, that is for quick debugging work, fix any of those with properly indented single # comments for your deliverable. This is the same thing as expecting a maintenance crew to clean up after themselves when finished.

You may work individually or in pairs at your option. Pairs will both submit the same thing in Sakai, only one hardcopy required.

The **deliverables** will be:

1) A .zip file with the zipped directory named in the form [lastName][Project6] (not a renamed .zip file) and the entire 06 directory inside, include all the N2T provided .asm files in their directories:

> Your project 06 directory should include your code and the provided test files. It should run and test properly from the command line. I will run the code via an automated script and should see your output match mine ***exactly***.

2) A hardcopy of your code. Each file should start on a new page, file header blocks should give all the appropriate identifying info.

3) A hardcopy of your grading sheet, name(s) filled in. This will make it's way back to you along with with any code annotations I make.