CS2002 P11 Addendum:

This addendum to the N2T book project description takes precedence where there is any duplication.

This project continues to work on the same python files as Project 10, to complete the high level Jack to VM compiler. The CompileEngine should now also generate a list of VM instructions (you can see below I called mine self.vmInstList). To avoid needless twiddling, also add this function you your CompileEngine:

```
    def get_vmInstructions(self):
        ''' returns a fully translated list of vm instructions, one instruction per list element '''
        return self.vmInstList
```

With this available your Analyzer needs minimal change to output the VM code file.

The test case files are in the 11 directory, just replace the project 10 test cases.

Create your new symbol table class as the book indicates, and the VMWriter can be done either as a class or just a module holding the indicated VM creation functions, and those functions only require one line each, maybe 2 lines in some styles of writing. But dead simple and no class data required.

The milestone check is a file line-by-line check, same as the final check was for Project 10.

The final tests are as per the book, runtime checks of operating programs in the VMEmulator.

Make sure you properly set up the `__main__()` for submission, both milestone and final.

There is an updated grade sheet for this project, do not use leftover sheets from CS2001.

> ***Code Quality*** is the instructors subjective judgement of the described attributes.
>
> ***Dry Test*** are the cold-hard silicon results of code known to you: All tests running without any crashes gains 20 points, each test that outputs a sub-comparisons successfully gains 10 more.
>
> As you can see above, on-time and broke is broke, and broke hurts. Late submission costs are 2 points for the first 24 hours and each day increases by 2 with a maximum late cost of 20 (2+4+6+8). Get working projects in! Don't turn in broke just to make on-time.

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.

NO Magic numbers!

2 person team submissions. You may ask each other teams questions, but no direct sharing of code. Chalk-talking through small sections is OK. The goal is for everyone to get comfortable and gain their basic competency. Providing "too much" assistance is not helpful to them as a grade is not competence.

> If a classmate is really stuck and it seems like help might get into the "too much" category, help them write the 60 second question and get it off to the instructor.

The **deliverable** is:

A. Milestone NLT 28 Nov

Submit your lastName11 directory zipped up to the milestone assignment in Sakai

B. Final submission NLT 13 Dec

1) *lastName*11 directory zipped up. No spaces in the name, it breaks directory searches and you don't want repair costs against your ***Well Built*** line.

2) A hardcopy of your code. Each file should start in a new page, file header blocks should give all the appropriate identifying info. Standard white background, it is impossible to leave feedback on black background printing.

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