CS2002 P7 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.)

Use the provided directory structure (with your last name laid in — no spaces!), it includes all the book-provided test cases and hook-up to my auto-testing program. Running the auto-test from the command line should generate

```
    End of script - Comparison ended successfully
```

five times indicating a successful run.

You should be running test cases in the CPUEmulator individually until all five pass, then run the auto test to make sure you properly set up the `__main__()` for submission.

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

> ***Well Built*** is the instructors subjective judgement of the described attributes.
>
> ***Working*** are the cold-hard silicon results: All tests running without any crashes gains 25 points, each test that outputs successfully gains 5 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!

Individual submissions. You may ask each other 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:

1) *lastName*07.zip

> 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.