



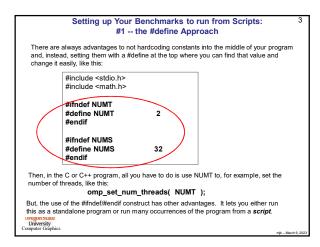
In this class, you are *required* to run your programs many times to observe the effect of different parameters on performance.

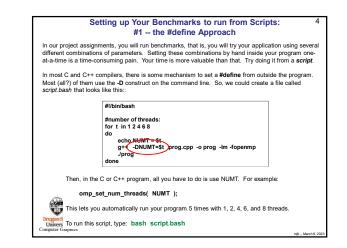
You could run those many versions one-at-a-time, but this could take hours. Or, you could write *scripts* that run all those parameter combinations in a couple of minutes.

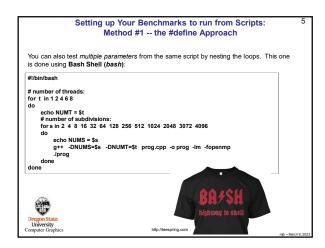
So, if you are the kind of person who has loads of free time on their hands and has nothing else they want to do, feel free to use the slow one-at-a-time approach.

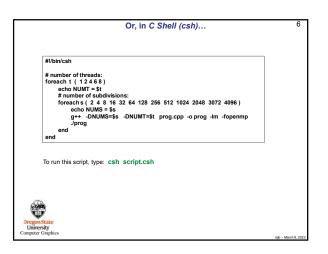
If you are not one of those, take a little time to learn how to do it via scripts.

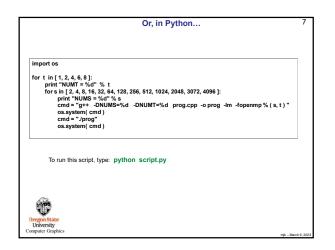


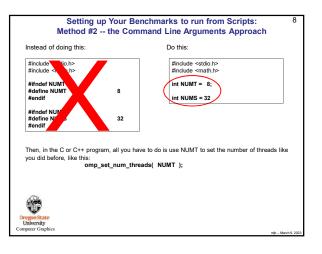


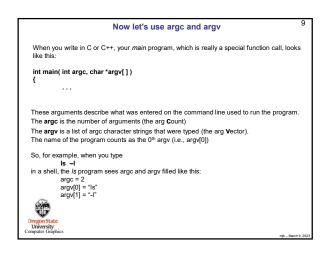


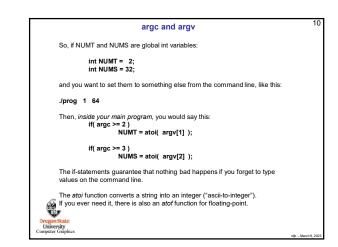


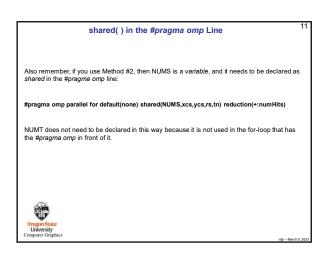


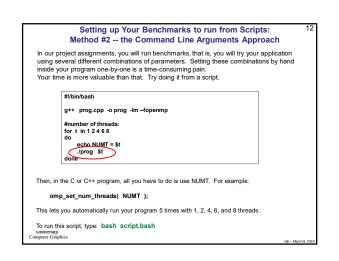


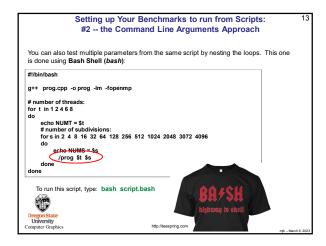




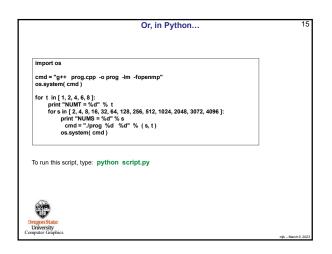












### Do Not Put These Loops in the .cpp Program!

### I know what you're thinking.

#### You're thinking:

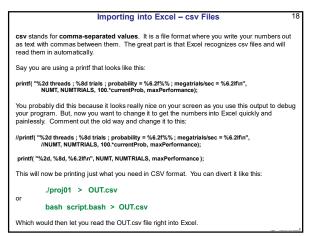
"Those scripts are different, and I've never done them before, and I don't want to take the 5 minutes to learn them. So, I'll just build the iterations through all the parameters into for-loops in the program."

Don't!

I see evidence that the first time OpenMP does anything, it also does some one-time setups. This will mess up your timing because your first test will seem slower than it should be and the others will seem artificially faster by comparison.

I recommend you run the program *separately* for each combination of parameters. (The script code in the previous pages shows that.)





### Diverting to a File

We all have a tendency to want to write our performance results out using *printf* (or *cout*) so that we can see them on the screen. That's fine. But, then we want to get those results into a file. You could mess with file I/O, or you could use a *divert* on the command line.

If you are currently running your program like this:

./proj01

and it prints to the standard output screen via printf or cout, then running it like this:

# ./proj01 > output.csv

will write your results into the file output.csv

(If you do it a second time, you will probably have to remove the previous  $\textit{output.csv}\xspace$  first.)

You can also divert the entire output (standard out and standard error) of a looping script:

## bash script.bash >& output

Dregon State University

