main.py

controls data flow between each module

* Starts with getComp() to get all components/system information and store in results
* Performs tests one at a time.
* Won’t run next module until it receives the data back from the module.
* Runs an internal timer to check how long the whole program took
* Create py UI which allows users to interact with the program and also displays timers for each test allowing the user to still see that its working.
* We will scrap the % done thing as it is very inconsistent and would be a lot of data sent between different files which could potentially slow cpu.
* Make the main program take as little as possible cpu performance.

GetComp.py

* get all components/system info from the pc and sends them back to main store in results

singleCPU.py

tests single core capability

* Do x number of calculations and return the speed which there done
* Check how large of calculations cpu can handle before it reaches x % cpu usage

memoryTest.py

* Run memory read/write test
* this can be done using psutil module
* create a score out of 1000 and return to main

disktest.py

* find out how many disks and drives pc has.
* Run test for each using psutil
* Create a score out of 1000 for the read/write speed
* Return score to main

multiCPU.py

test multicore capabilities

* Run x number of calculations on all cpu cores at once and returns to time for all of them. Then finds average of all times.

writeCSV.py

takes data from a dictionary and stores in a 1D array in a text file and sends this

* Data should be stores as small as possible to allow fast transfer
* Should be encrypted
* Tells sendCSV to send the CSV to the server
* Deletes the CSV file to save space

sendCSV.py

* sends CSV file to server to be read by website and stored in a database