**Initial Test Features (i.e., Python tests)**

* Run a test and return a SUCCESS/FAIL result.
* Every test must implement a run() method. Determine the run() parameters and return values, which should be consistent across all tests.
* Used to tests computer’s major subsystems (need of error handling).

1. **Memory Test -** An example of a test to exercise memory is a Python program that creates a list with one million items. Use exception handling to determine if there is an error and print "ERROR:" with the error text. On success print "SUCCESS".

The memory test program is a Python program that creates a one megabyte list and

fills the list with consecutive integers. It does this five consecutive times

in a loop. There are three main scenarios to test:

The success scenario prints: SUCCESS

The program also has a feature to determine the length of time for creating the lists.

If the duration is longer than some number of seconds, the program prints an error message beginning

with ERROR. The calling program should always check for "ERROR".

If an exception is thrown (e.g., Memory) the program prints "ERROR" and the exception message.  
All errors should begin with the text "ERROR:"

=======================

Test Steps

=======================

1) SUCCESS scenario

Run the test configured for five iterations of creating a one megabyte list

Expected result: Prints SUCCESS

2) Time exceeded scenario - This test makes sure the program handles

the scenario where the runtime of the test exceeds a threshold.

Add a line to the code to force the program to exceed the time limit For example:

add a call to the time.sleep() function to add 5 seconds to the run time.

Expected result: Should print ERROR - Memory test exceeds time boundary

3) Memory and other exceptions scenario

This test ensures the program handles exceptions properly. Add code to the program to

force an exception. For example, force the program to access a memory list beyond a

boundary. memList[listCount + 1]

Expected result: Should print: ERROR - list index out of range

1. **Storage** - An example of a test that exercises the storage system is a Python program that creates a one megabyte file with random characters. Use exception handling to determine if there is an error and print "ERROR:" and the error text. On success print "SUCCESS".

The storage test program is a Python program that creates 3mb files (i.g. with 3M consecutive integers), then it reads and writes data from the file to test the storage system. This is done 100 times in a loop.

**Feature: times the transfer time between the read and write speed to measure performance** (if done when creating 1 time at a time in the loop versus 10 files at a time?)

Get time stamp

Loop 50 times

1. Create file
2. Wrte 3MB integer
3. Close file
4. Open file
5. Read file

Get time stamp

Except: if too long between both stamps

Plus other error handling errors

=======================

Test Steps

=======================

1) SUCCESS scenario

Creates files, reads and writes

Expected result: Prints SUCCESS

**2) Time exceeded scenario** – Tests the time sure the program handles the scenario where file generation goes over the storage capacity.  
Add a line to the code to force the program to exceed the storage limit.  
Expected result: Should print ERROR - Storage test exceeds boundary on time

3) Storage and other exceptions scenario

This test ensures the program handles exceptions properly. Add code to the program to

force an exception.   
For example, **force the program to access a file not created.**

Expected result: Should print: ERROR - file does not exist.

1. **Networking** - An example of a test that exercises networking is a Python program that connects to a website, downloads an HTML file and verifies the contents. On success print "SUCCESS", otherwise print "ERROR" and the error text.

open http, connects to wesite, open up a file and search for some strings on it.

The networking test program is a Python program that connects to a website and HTML file we created and verifies its contents. Here, the test looks for specific scripts in the file (such as the heading) to verify a network connection is set. This is done for **1** **website.**

=======================

Test Steps

=======================

1) SUCCESS scenario

Connects to a website, downloads an HTML file and verifies the contents  
Expected result: Prints SUCCESS

2) Network and other exceptions scenario

This test ensures the program handles exceptions properly. Add code to the program to

force an exception.   
**For example, force the program to access an non-existing html website and …?**

**Expected result: Should print: ERROR: website does not exist / network configuration is wrong?**

1. **CPU** - An example of a CPU test is a program that generates one million random numbers. On success print "SUCCESS", otherwise print "ERROR" and the error text.

A list

The CPU test program is a Python program that creates a list…..

Feature: ?

=======================

Test Steps

=======================

1) SUCCESS scenario

Expected result: Prints SUCCESS

**2)**

3) Storage and other exceptions scenario

This test ensures the program handles exceptions properly. Add code to the program to

force an exception.   
For example,

Expected result: Should print: ERROR: file does not exist.

1. Video - An example of a Python program that tests the video subsystem is program that uses the pyscreenshot module to create a PNG file screenshot. The program should verify the file is created and has a > 0 file size. On success print "SUCCESS", otherwise print "ERROR" and the error text.
2. Integer Math Test - Measures how fast the CPU can perform integer operations. An integer is a whole number with no fractional part. On success print "SUCCESS", otherwise print "ERROR" and the error test.
3. Prime Number Test - Measures how fast the CPU can search for prime numbers. example, 1, 2, 3, 5, 7, 11, etc. This algorithm should use a loop. On success print "SUCCESS", otherwise print "ERROR" and the error text.
4. Floating Point Test - Performs the same operations as the integer math tests, however with floating point numbers. A floating point number is a number with fractional parts (e.g., 10.12345). On success print "SUCCESS", otherwise print "ERROR" and the error text.