Approach to Problem

# Initial Development Goals (Project 2):

1. Confirm the ability to open a .dll file.
2. Read/use the contents of the .dll.
3. Use variable names for the .dll filename and test functions
4. Enqueue a .dll filename and test to a queue (hard coded for test purposes)
5. Dequeue a .dll filename and test from a queue to be used in the program.
6. Test Harness is the driver for this iteration.
7. Pass a pointer to child test object (only one object, eventually this will be a thread object).
8. Pass the results of the test back to the harness object.
9. Pass information to a logging queue.
10. Do some initial error testing.

# Additional Development Goals (Project 3):

1. Everything in Project 2 above.
2. Test Harness running as the server.
   1. Child Test objects implemented as function in Test Harness (multi-threaded).
3. Test Executive running as the client.
4. Graphic User Interface (GUI) using MFC.
5. Able to select multiple DLLs and multiple tests. Selective testing.
6. Implement socket communication.
7. Data passed as binary data from the GUI client to Test Harness.
8. Blocking queues (used for message queues and logging queues).
9. Multi-threading.
10. Thread Pool.
11. Concurrency Synchronization.

# Additional Development Goals (Project 4):

1. Everything in Project 3 above.
2. Code Cleanup
3. Logger changes
4. Exception message change
5. Comments
6. Two instructor demos

# RAA

## Mudit Vats

1. Setup and manage GitHub.
2. Initial .dll development effort to open/close and read .dll contents.
3. Pointer to function.
4. Initial tests using TestfuncA DLL.
5. GUI development.
6. Integration of TestExec and TestHarness

## Troy Archer

1. Child Test Class development.
2. Logger class development

## David Howick

1. Setup initial skeleton of classes.
2. Continued work with .dll to use variables for filenames and test function names.
3. Test Harness class development.
4. Initial multi-threading development.
5. Thread pool and synchronization efforts.

## David Pretola

1. Message class development.
2. Template class development for queues.
3. Blocking queue development.
4. Communication Sockets
5. Serialization (using binary data)

# Github URL

* Project Location: <https://github.com/KodeFu/cse687>
* Git Clone URL: <https://github.com/KodeFu/cse687.git>

Please see the README.md file for build instructions and additional notes.