

ED1 Thread Pool

# Overview | Objective

This lab is designed to show you a simple implementation of a thread pool system. It will test your familiarity with pointers, multi threading, mutexes, working within a pre-existing code base, and problem solving skills. Be sure to read all related .h / .cpp, read through the slides, google when you get stuck on specific functions.

Comment out the calls to our solution version and write your own implementation.

# Pool Usage

To use the thread pool in the FSGDEngine follow the following steps

1. Toggle the input console with **~**
2. Invoke the thread pool with **PoolTask 12345**
3. Check the console output window for results

# Code Section

You will need to complete the following items in the EDTheadPool project, see grading breakdown for each individual function

## ThreadPool.cpp

All of the functions in the TheadPool class must be implemented

# Rubric | Grading Breakdown

Commenting should be appropriate and used where necessary.

Your completed lab should be warning free.

Your completed lab should be error/crash free, a submission that crashes will result in a 0%.

Your project should be free of memory leaks.

Failure to follow directions will result in the loss of additional points.

|  |  |
| --- | --- |
| **ThreadPool** |  |
| Init | 10% |
| PostWork | 10% |
| GetWork | 15% |
| ClearWork | 10% |
| Execute | 20% |
| Shutdown | 20% |
| **Capstone** |  |
| Use thread pool | 15% |

# Hints

* The functions in the grading breakdown are listed in the order you should probably write them
* Don’t forget to lock all shared resources
* Don’t forget to check pointers and container sizes
* Don’t forget to deallocate all resources
* Don’t forget to read the documentation for functions you are calling, I specifically did not go over every detail because I want you to look them up
* Check the memory manager interface to see how to create a heap, 1MB should be fine, be sure to give the heap a unique name per thread
* For capstone, put something interesting on a thread, it can be something existing or it can be something that you add, it can be invoked from the input console or something that is runtime

# Submissions

Lab is due at **the end** of the lab period. Have a lab instructor grade your assignment; you still need to turn in the assignment on VFILER, grades will be posted on LMS. If you are off campus, late assignments can be turned in through LMS.

This lab must be turned in using the .zip file format using as *LastName.FirstName.lab\_name.zip*.

To create the zip file, run the “make submission folder.bat” file. It will create a folder one level higher called “zip\_this\_and\_turn\_in,” leave it named that. Zip it, rename the zip file and turn it in on VFILER.