Gam 475/SE 485
Spring 2023

use Adobe Reader to complete

(Type in fields)

Submission Report
Keenan

PA4 – File

Student Information

Integrity Policy: All university integrity and class syllabus policies have been followed. I have neither given, nor received, nor have I tolerated others' use of unauthorized aid.

I understand and followed these policies: Yes   No

Name:

Date:

Submission Details

Final **Changelist** number:

Verified build: Yes   No

Number Tests Passed:

Required Configurations:

Discussion (What did you learn):

Gam 475/SE 485
Spring 2023

use Adobe Reader to complete

(Type in fields)

Submission Report
Keenan

Verify Builds

- Follow the Piazza procedure on submission
  - Verify your submission compiles and works at the changelist number.
- Verify that only MINIMUM files are submitted
  - No – Generated files
    - *.pdb, *.suo, *.sdf, *.user, *.obj, *.exe, *.log, *.pdb, *.db, *.user
    - Anything that is generated by the compiler should not be included
  - No – Generated directories
    - /Debug, /Release, /Log, /ipch, /.vs
- Typical files project files that are required
  - *.sln, *.cpp, *.h
  - *.vcxproj, *.vcxproj.filters, CleanMe.bat

Standard Rules

Submit multiple times to Perforce
- Submit your work as you go to perforce several times (at least 5)
  - As soon as you get something working, submit to perforce
  - Have reasonable check-in comments
    - Points will be deducted if minimum is not reached

Write all programs in cross-platform C++
- Optimize for execution speed and robustness
- Working code doesn't mean full credit

Submission Report
- Fill out the submission Report
  - No report, no grade

Code and project needs to compile and run
- Make sure that your program compiles and runs
  - Warning level ALL …
  - NO Warnings or ERRORS
    - Your code should be squeaky clean.
  - Code needs to work "as-is".
    - No modifications to files or deleting files necessary to compile or run.
  - All your code must compile from perforce with no modifications.
    - Otherwise it's a 0, no exceptions

Project needs to run to completion
- If it crashes for any reason…
  - It will not be graded and you get a 0

Gam 475/SE 485
Spring 2023

use Adobe Reader to complete

(Type in fields)

Submission Report
Keenan

No Containers
- NO STL allowed {Vector, Lists, Sets, etc...}
  - No automatic containers or arrays
  - You need to do this the old fashion way - *YOU EARNED IT*

Leave Project Settings
- Do NOT change the project or warning level
  - Any changing of level or suppression of warnings is an integrity issue

Simple C++
- No modern C++
  - No Lambdas, Autos, templates, etc…
  - No Boost
- NO Streams
  - Used fopen, fread, fwrite…
- No code in MACROS
  - Code needs to be in cpp files to see and debug it easy
- *Exception:*
  - implicit problem needs templates

Leaking Memory
- If the program leaks memory
  - There is a deduction of 20% of grade
- If a class creates an object using new/malloc
  - It is responsible for its deletion
- Any *MEMORY* dynamically allocated that isn't freed up is *LEAKING*
  - Leaking is *HORRIBLE*, so you lose points

No Debug code or files disabled
- Make sure the program is returned to the original state
  - If you added debug code, please return to original state
- If you disabled file, you need to re-enable the files
  - All files must be active to get credit.
  - Better to lose points for unit tests than to disable and lose all points

No Adding files to this project
- This project will work "as-is" do not add files…
- Grading system will overwrite project settings and will ignore any student's added files and will returned program to the original state

UnitTestConfiguration file (if provided) needs to be set by user
- Grading will be on the UnitTestConfiguration settings
  - Please explicitly set which tests you want graded… no regrading if set incorrectly

Gam 475/SE 485
Spring 2023

use Adobe Reader to complete

(Type in fields)

Submission Report
Keenan

Due Dates

- See Piazza for due date and time
- Submit program perforce in your student directory assignment supplied.
- Fill out your this ***Submission Report*** and commit to perforce
  - ***ONLY*** use Adobe Reader to fill out form, all others will be rejected.
  - Fill out the form and discussion for full credit.

Goals

- Create a File System
  - Basic File system

Assignments

1. **Create the Memory program in C++, for Visual Studio 2022**
   - Document the code
   - Code should be Warning Level ALL
     - It doesn't compile -→ 0 for grade
   - Compiles in both configurations
     - Debug x86
     - Release x86

2. **Create a complete File System**
   - Win32 library
   - Document the code
   - Code should be warning free

3. **File System**
   - Files system should be able to:
     - Open / Close file
     - Read / Write file
     - Flush – flush pending write buffers to the file
     - Seek - move to a point in the file
     - Handle Based – No direct use of file pointers exposed to the user of the system
   - Suggested implementation
     - Wrap Win32 file system calls.
       - CreateFile(), ReadFile(), WriteFile(), CloseHandle()….
     - Create an interface that is easy and intuitive to use
       - Remember your interface does not need to be 1:1 with win32

Gam 475/SE 485
Spring 2023

use Adobe Reader to complete

(Type in fields)

Submission Report
Keenan

- - You need to make decisions on how to wrap the functions
    - Assume all file system calls are synchronous
  - o File system debugging
    - Make sure you can create, read, write, seek, and flush arbitrary files.
    - Make sure you can read different types of data, nothing hard coded

4. **Suggested User case programs to help you out, but only the Unit functions will be graded**
   - o Implement individual demos that exercise the complete interface
     - Open, Close, Read, Write, Seek, Tell, Flush
   - o Exercise performance, where you need to flush data
     - Data is generated inside a for loop
     - Flushing after each write
   - o Load a series of data structures
     - Create a series of data structures in code
     - Write them off to a file
     - Read them back from a file
     - Compare the data

Validation

Simple checklist to make sure that everything is submitted correctly

- Is the project compiling and running without any errors or warnings?
- Does the project run **_ALL_** the unit tests execute without crashing?
- Is the submission report filled in and submitted to perforce?
- Follow the verification process for perforce
  - o Is all the code there and compiles "as-is"?
  - o No extra files
- Is the project leaking memory?

Hints

Most assignments will have hints in a section like this.

- Write your API first and try to write your test code first.
  - o Incremental development!
- Coding isn't the hard part of this assignment.
  - o Planning and design is the hardest.