Carnegie Mellon University

24783: Advanced Engineering Computation

Monday and Wednesday, 4:30PM-6:20PM, On Zoom

Semester: Spring**, Year:** 2021

Units: 12, Section(s): A

Instructor information

Name	Dr. Soji Yamakawa
Contact Info	soji@andrew.cmu.edu
Office hour location	Online
Office hours	Friday 2100-2200

TA Information [If applicable]

TA name	Joe Joseph
TA Contact Info	joej@andrew.cmu.edu
Office location	TBA
Office hours	TBA

TA name	Pranshu Pant
TA Contact Info	ppant@andrew.cmu.edu
Office location	TBA
Office hours	TBA

Course Description

This course covers the advanced programming and computational skills necessary for solving engineering problems. These include (1) efficient data structures and algorithms for modeling and processing real-world data sets such as trees, hash tables, searching, priority queues, etc. (2) techniques for simulation and visualization such as numerically solving ODEs and PDEs, viewing control, programmable shader, etc., (4) tools for version controlling, scripting, and code building including sub-version, git, and cmake, and (5) cross-platform and long-term software development. Students will experience practical training in the above knowledge and programming skills through bi-weekly assignments and a final team project.

Prerequisite: 24-780 Engineering Computation or equivalent programming experience.

Learning Objectives

C++ Programming

- Cross-Platform and Long-Term software development,
- Creating a project for Microsoft Visual C++ and Apple XCode using CMake,
- Managing source code with Git,
- Writing, debugging and compiling a code in C++.

Engineering Computation

- Selecting and/or designing an algorithm, and data structure for engineering applications,
- Understanding the new trends in computation,
- Visualizing with OpenGL and programmable shader,
- Writing re-usable code,
- Cross-platform and long-term software development,
- Coding for portable devices.

Learning Resources

Textbooks will help better understand the course topics, but not required.

- The C++ Programming Language: Fourth Edition, by Bjarne Stroustrup, ISBN-13: 978-0321563842
- The C Programming Language (2nd Edition), by Brian W. Kernighan and Dennis M. Ritche, ISBN 978-0131103627
- OpenGL programming Guide: The Official Guide to Learning OpenGL Version 2, 5th Edition, by OpenGL ARB, Dave Shreiner, Mason Woo, Jackie Neider, and Tom Davis, ISBN 978-0321335739

Assessments

The final course grade will be calculated using the following categories:

Assessment	Percentage of Final Grade
9 assignments	8%x9=72%
Project	28%

- Assignments are given every two weeks (except the first three). Regular schedule is, a Problem Set is uploaded to the Blackboard on Monday and due a week after next Monday noon. Grade and solutions are posted in 14 days of submission due date.
  Please strictly adhere to the filename and project-name and directory-structure guideline when you prepare your submission. Teaching assistants grade assignments with an automated script. If your filename does not follow the guideline, the script stops. (You really don't want to be graded by a frustrated TA!)
  For assignments, you can study in a group. I encourage you to do so. However, each of you needs to write your solution independently after understanding the concept. In most assignments, you can cut & paste code from the sample code presented in class. However, NEVER copy & paste code from your friend's solution.
- In the final project, students write a program in 3- to 5-student team. There are two project options:
  - Option A: Find an engineering problem and solve it by C++ programming. It can be a problem that you face daily basis. It can be related to your research work. For example, are there manual steps you are doing over and over again? Many of such processes can be automated by programming.
  - Option B: Port a classic arcade game from around 1990. Pick one arcade game from around 1990, and port it. The graphics and audio do not have to be exact. But the program must replicate essence of the game. If the game play continues more than five minutes, you need to implement at least first five minutes into the game play.

Students will be assigned the following final letter grades, based on calculations coming from the course assessment section.

Grade	Percentage Interval
A	95% and higher
A-	90-94.99%
B+	87-89.99%
B	84-86.99%
B-	80-83.99%
C+	77-79.99%
C	74-76.99%
C-	70-73.99%

D+	65-69.99%
D	60-64.99%
R (F)	Below 60%

Grading Policies

- **Late Policy:** Everyone has 3 late tickets. If you miss the deadline but submit the assignment within 24 hours of deadline, you use 1 late ticket with no late penalty until you run out of your late ticket. Once you run out of late ticket and miss the deadline or if you miss by more than 24 hours, no point is given. Extensions will be granted for acceptable reasons, including medical condition, academic conferences, family emergency, etc. Your best friend's wedding counts as an acceptable reason. However, your own vacation doesn't.
- **Re-grade policy**: Please review comments from TAs when the grade is posted and make sure there is no error in grading. If you find a grading error, you need to let the instructor know as soon as possible but no later than a week from the date your grade is posted. The grade may not be corrected after one week.
- **Missed Quizzes**: A missed quiz counts as zero credit unless you get permission in advance from the instructor. If you are sick a note from your doctor or the student health center is required. A make-up quiz may not be of the same difficulty as the in-class quiz. The instructor can also give an oral make-up quiz instead.

Course Policies

- **Academic Integrity & Collaboration**: Any act of cheating or plagiarism will be treated in accordance with Carnegie Mellon's Policy on Academic Integrity, which can be found on https://www.cmu.edu/academic-integrity/.
- **Accommodations for students with disabilities**: If you have a disability and require accommodations, please contact Catherine Getchell, Director of Disability Resources, 412-268-6121, getchell@cmu.edu. If you have an accommodations letter from the Disability Resources office, I encourage you to discuss your accommodations and needs with me as early in the semester as possible. I will work with you to ensure that accommodations are provided as appropriate.
- **Statement on student wellness**: As a student, you may experience a range of challenges that can interfere with learning, such as strained relationships, increased anxiety, substance use, feeling down, difficulty concentrating and/or lack of motivation. These mental health concerns or stressful events may diminish your academic performance and/or reduce your ability to participate in daily activities. CMU services are available, and treatment does work. You can learn more about confidential mental health services available on campus at: http://www.cmu.edu/counseling/. Support is always available (24/7) from Counseling and Psychological Services: 412-268-2922.
- **Mobile Devices**: Students are allowed to take notes of take pictures of the contents shown on the screen or on the white board. All mobile devices must be set to the silent mode during the class.

Course Schedule

Lecture	Topic	Problem Set
1	Quick Recap from 24-780 Using Terminal (Command Line)	PS1 Creating an account on the course Git

	CMake Version Control System	repository Compiling with CMake
2	Compiling with Imported CMake Library Projects Concept of Event-Driven Programming	
3	More about Terminal Converting a polling-based style to an event-driven style	PS2 Converting a polling-based program to an event-driven program
4	Review template, inheritance, resource management (writing a Bitmap class)	
5	PNG Bitmap Viewer Concept of “Move” Command Parameters	PS3 Converting a polling-based program to an event-driven program
6	R-value reference Move-assignment operator and move-constructor	
7	Hash Set and Hash Table	PS4 Finishing SimpleBitmap class Pattern matching
8	Binary Tree Construction	
9	Making a better-protected binary-tree class Visualizing a binary tree	
10	Binary-tree manipulations - Deletion - Rotation	
11	3D Graphics Modern 3D-graphics APIs Vertex-attribute arrays View control Lighting	PS5 Tree re-balancing (1) Stout and Warren’s method (2) AVL-tree
12	3D Graphics View control Lighting Binary STL	
13	CPU Endian-ness Polygonal-mesh data structure Picking	
14	Adding topological inquiry in the polygonal-mesh data structure	

15	Faster intersection checking with a lattice	PS6 Generating a geometric model of a NACA airfoils
16	Finding connected vertices Finding neighboring polygons	
17	Color-map problem Depth-first and breadth-first search	
18	Greedy-search path finding Flood-fill in a polygonal mesh	
19	Edge-collapsing operator Priority-queue that allows value updated	PS7 Mesh painting, simulated laser-range scanner, and slices
20	Mesh simplification with quadric error metric Installing Resources with CMake	
21	Programmable Shader Vertex shader and fragment shader OpenGL Shader Language (GLSL)	
22	Uniform Color Generic Attribute Texture (sampler2D)	
23	Texture (sampler2D) Plain 3D renderer Rainbow color based on the vertex position Gouraud shading Phong shading	PS8 Mesh segmentation and boundary extraction
24	Gouraud shading Phong shading Billboarding Point sprite	
25	Point sprite Particle method	
26	Coding for Portable Devices	PS9 Drawing a cubic-bezier surface using GLSL programs
27	Coding for Portable Devices	
	Final Presentations	