

CS3331 Concurrent Computing

Course Information

*It takes a really bad school to ruin a good student
and
a really fantastic school to rescue a bad student.*

1

Spring 2021

Dennis J. Frailey

General Information: 1/5

- **Course:** CS3331 Concurrent Computing
- **Office:** Rekhi 305
- **Instructor:** Ching-Kuang Shene aka **C-K**
- **Meeting:** M/W/F 2-2:50pm **Remote Async**
- **Office Hrs:** No F2F Meeting. Use email or Zoom meeting
- **Textbook:** No textbook; but you will have my more than 500+ slides and web-based materials.
- **Exams:** *two* exams and *one* final
- **Programming:** *six* programming assignments
- **Others:** Many Questions will be posted in the videos and on Weekly Reading Lists
- Check the course page frequently for announcements and weekly reading
- **No late programs will be graded**

General Information: 2/5

- www.csl.mtu.edu/cs3331.ck/www/Home.html
- [/classes/cs3331.ck/common](http://classes.cs3331.ck/common) **or**
[/local/cs3331.ck/common](http://local.cs3331.ck/common) **or**
[/mtu/cs3331.ck/common](http://mtu.cs3331.ck/common) **will have all slides used in class, software, etc. Check it frequently. More Later!**
- **This is a programming intensive class.** Always start working on programming assignments ***EARLY!*** Except for a valid excuse with proofs, no extension will be given.
- We will use C and later C++. C++ will be reviewed to fit our need but won't go very deep.
- I do not have an attendance policy. However, ***if you fail, you fail.*** ☹
- Week 5 to week 11 cover the most difficult topics.

General Information: 3/5

- Use the command line version of `submit` and verify that all required files have been submitted.
- If you use the command line version of `submit`, use the `recover` command to check for submitted files.
- Use `dos2unix` or similar utilities to convert your Windows test files to Unix before submission.
- **WE ONLY ACCEPT TEXT FILES. DO NOT FORMAT YOUR FILES WITH A WORD PROCESSOR.**
- Unix filenames are **CASE SENSITIVE!**

General Information: 4/5

- The following is **VERY** important to remember:
 - ❖ My grading policy is based on how many key points you have successfully answered.
 - ❖ If a problem has four key points and is assigned 20 points, you get
 - 0 point if all four key points are missing
 - 5 points if you get one key point right
 - 10 points if you get two key points right
 - 15 points if you get three key points right
 - 20 points if you get all key points right.
- **I do not do grade inflation!**

General Information: 5/5

■ Course Outline

- ❖ Part 1: Introduction

- ❖ Part 2: Processes and Threads

- ❖ Part 3: Synchronization (**Most difficult. Don't skip classes**)

 - Mutual Exclusion and Its Solutions

 - Locks, Semaphores, Monitors, Condition Variables

 - Race conditions, Deadlocks and Livelocks, Busy Waiting and Starvation

 - Simple Message Passing

- ❖ Part 4: Programming Interfaces and Language Supports

- ❖ Part 5: Concurrent Architectures and GPU Programming

FAQ: 1/5

- The following is **VERY** important to remember:
 - ❖ My grading policy is based on how many key points you have successfully answered.
 - ❖ You may use your wording to answer a question. However, you have to make sure every point will be addressed properly.
 - ❖ Thus, **grading is not random and is fair.** It is a correctness-based approach. Hence, it is fair and is not a way to make you feeling fair.
- **I do not do grade inflation! You get what you have answered correctly.**

FAQ: 2/5

- **Here is an example:**
 - If a problem has four key points and is assigned 20 points, you get
 - ✓ 0 point if all four key points are missing
 - ✓ 5 points if you get one key point right
 - ✓ 10 points if you get two key points right
 - ✓ 15 points if you get three key points right
 - ✓ 20 points if you get all key points right.
- **Note that what you think is correct may not be considered correct by me. Read the slides carefully.**
- **Again, I do not do grade inflation!**

FAQ: 3/5

- **Each programming assignment has its own grading sheet.**
- **You may find this blank grading sheet near the bottom on the assignment page when it is online.**
- **In this way, you know what the key elements are for each assignment.**
- **Always read the submission guidelines carefully and make sure your program is correct.**

FAQ: 4/5

- You may ask for a regrading for each exam and program.
- Regrades must be requested within 7 days from the day you get the quiz, exam or program back. Regrade requests made out of this time frame will not be regraded and be returned immediately.
- You have to write clearly why a regrade is needed. Note that this writing must be correctness-based.
- Allow for at least seven days or the next grade post to see an upgrade. **If your grade is not updated, it means your grade does not change.**
- You are responsible to pick up your regraded stuffs either by email or postal mail.

FAQ: 5/5

- You must receive 60% in both exam and programming to have a pass grade.
- What does this 60% mean?
 - I have a record of the performance of all students in the past 10 years.
 - This 60% is calculated based on this database rather than a particular class. The purpose is to make sure all students from this class will meet the average of **ALL** previous students.
 - Sometime around the 6th week, those who are likely to fail will have a * shown on each category (i.e., programming and exam).

Other Stuff: 1/4

- We will use asynchronous remote lectures. In this way, you have the freedom to study at any time.
- You have to do a good time management and the willingness to do your work.
- Videos will be pre-recorded and available in the common directory
`/classes/cs3331.ck/common`
and on the class slide page.
- Each week a Weekly Reading List is available in which you have working problems, videos and programming assignments.

Other Stuffs: 2/4

- The slide page will be updated with new slides and videos. So, check this page frequently so that you will be up-to-date.
- If you have a **Linux** system, you may use it to do ***ALL*** assignments.
- If you have a macOS system, you may use it to do the first two assignments.
- If you use Windows, you may use PuTTY to ssh to a CS server such as colossus. WinSCP is a good software for transferring files from your Windows to your CS Linux account.

Other Stuff: 3/4

- The first Programming Assignments are Unix/Linux based. You may use either macOS or Linux locally and transfer the files to be submitted to your CS account.
- If you use a Windows system, you must use a `ssh` capable system to login to a CS machine to do your work. `PuTTY` is a good choice.
- The last four programming assignments use a software **ThreadMentor** developed under the support of National Science Foundation.
- The statically linked Linux 64-bit version is in the `common` directory.

Other Stuff: 4/4

- If you have a Linux 64-bit system, you can install **ThreadMentor** directly.
- If you use macOS or Windows, you may download VMware Fusion (macOS) or VMware Workstation (Windows). Then, install a Red Hat Linux on this virtual machine and put **ThreadMentor** on this Linux system.
- User guides should be available soon!
- **IMPORTANT:** The system load on your system is much lighter than on a CS server. You must test your programs on a CS machine, because we use a CS machine for grading.

Website: 1/6

CS3331 Concurrent Computing Course Information Page Fall 2020



Course Instructor: Dr. C.-K. Shene
Office: Rekhi 305
Class Meeting: Monday and Wednesday 16:05 - 17:20
Classroom: R&M U115
Office Hour: Monday and Wednesday 15:05 - 15:55
Phone: (906) 487-3392
e-mail: shene@mtu.edu

<http://www.csl.mtu.edu/cs3331.ck/www/Home.html>

[Click here](#) to see the most recent announcements (December 4, 2020)
[Click here](#) to see the most recent ThreadMentor FAQ (March 8, 2019)
[Click here](#) for the most recent grade posting (December 22, 2020)

Some Important Advises

- This course will use C and C++. We will discuss C++ to some degree barely enough for our purpose. Thus, you should get a good reference of C/C++ in hand.
- Always start doing your programming assignment **EARLY**. Waiting until the last few hours is definitely not a wise move, because there could be too many people running their programs at the same time making system response time extremely slow. Moreover, machines may crash causing extra delay. Normally, I do not grant any extension if your home machine, your phone line or the department machines crash in the last minutes. **Always start early.**
- Concurrent programming requires a careful planning, which may need a mindset that is different from that of sequential programming you learned in previous courses. When you receive a programming assignment, **do not sit in front of a workstation and start typing your program immediately.** This trial-and-error approach, in general, **DOES NOT** work and may require longer time to finish your assignment. The best way of doing concurrent programming assignments is to think carefully with a pencil and a piece of paper **before** touching a computer keyboard.
- Since the behavior of a concurrent program is dynamic, which means it acts differently from time to time, you should try to reason about the correctness of your program. In general, just running for one or two test cases is not sufficient. Our grader will read your program to find possible errors. When writing sequential programs, the same bug appears at the same place every time you run the same program. However, bugs of a concurrent program (e.g., race conditions and deadlocks) may not appear every time. As a result, you may have an incorrect program even though you may have had hundreds correct test runs.
- **Due to security reasons, starting with Fall of 2008, you will not receive submission confirmation e-mails.** As a result, you should use the interactive version of submit and ensure all required files are submitted properly. If you choose to use the command line version of submit, you should use the recover command to verify your submission.
- After submitting your assignment, **DO NOT** modify your files so that the system administrator can retrieve your files in case your submission did not go through successfully.
- Always read the assigned materials listed in the weekly reading lists and do the problems. Weekly reading lists are available near the bottom of this page.

Catalog Description

This course will discuss concepts and techniques in concurrent computing. Major topics include: processes and threads, mutual exclusion, semaphores, monitors and condition synchronization, deadlock, safety and liveness, message passing, and concurrent architectures.

Course Credits, etc.

Credits: 3.0, Lec-Rec-Lab: (0-3-0), Semesters Offered: Fall, Spring

Course Prerequisites

CS 1142 or (CS1141 and CS1040) and CS2311 and CS2321

Course Objectives and Learning Outcomes

1. Know the basics of concurrent computing

Website: 2/6

Topics Covered

1. Basic systems concepts important to concurrent computing
2. Multi-process and multithreaded programming basics
3. Race conditions, critical sections, and synchronization
4. Pure software and hardware solutions and proof techniques for mutual exclusion
5. Synchronization primitives (i.e., semaphores, mutex locks, monitors of Hoare and Mesa types, basic message passing)
6. Deadlocks and livelocks
7. Survey of some languages (e.g., Java and Ada) and libraries (e.g., Pthreads)

Textbook

There is no suitable textbook for this course. All slides used in class and supporting materials are available:

- in the common directory `/classes/cs3331.ck/common` or `/local/classes/cs3331.ck/common` or `/mtu/classes/cs3331.ck/common`
- [this page](#) if you prefer web access.
- Software tools and example programs are only available in the common directory.

Check the `/classes/cs3331.ck/common` directory or [this page](#) frequently for new slides and other available stuffs.

Other Useful Materials

- [Make your program](#)
- [Multithreaded Programming with ThreadMentor](#)
- [Solaris multithreaded programming](#)
- [Unix Multiprocess Programming](#)
- [Signals](#)
- [Non-Local GOTO: setjmp\(\) and longjmp\(\)](#)

```
cd /classes/cs3331.ck/common
```

Meeting Time and Office Hours

| | Monday | Tuesday | Wednesday | Thursday | Friday |
|---------------|--------------------------|---------------|--------------------------|-------------|-----------------------|
| 13:05 - 13:55 | | Dept. Meeting | | TPR Meeting | |
| 14:05-14:55 | CS5631 Rekhi G009 | | CS5631 Rekhi G009 | | CS5631 Rekhi G009 |
| 15:05-15:55 | Office Hour Rekhi 305 | | Office Hour Rekhi 305 | | Department Seminar |
| 16:05-17:20 | CS3331 M&M U115 | | CS3331 M&M U115 | | |

After class and e-mail discussions are welcome
M&M: Minerals & Materials Engineering Building

EEWindowEvaluati...pdf

Show all

Sheet1

100%

Website: 3/6

Important Dates

The following is a list of important dates. The exam dates are fixed and will not change; however, programming assignment due dates may be modified based on the course pace. Changes will be announced in class and on the [info](#) page.

important dates

| | | | | |
|------------|--------------------|----------------------|--------------------|------------------------|
| Week 4 | September 25 | Friday | 11pm | Program 1 due |
| Week 6 | October 5 and 7 | Monday and Wednesday | in class | Exam 1 (50 min) |
| Week 7 | October 16 | Friday | 11pm | Program 2 due |
| Week 9 | October 30 | Friday | 11pm | Program 3 due |
| Week 10 | November 6 and 4 | Monday and Wednesday | in class | Exam 2 (50 min) |
| Week 12 | November 20 | Friday | 11pm | Program 4 due |
| Week 13 | December 4 | Friday | 11pm | Program 5 due |
| Week 14 | December 11 | Friday | 11pm | Program 6 due |
| Final Week | December 14 | Monday | 5:30-7:30pm | Final (2 hours) |

**Date/Time changes will be announced in class.
Usually, I will follow the schedule shown above.**

Weights Assigned to Programs

We will have maximum six programming assignments.

| Program 1 | Program 2 | Program 3 | Program 4 | Program 5 | Program 6 |
|-----------|-----------|-----------|-----------|-----------|-----------|
| 50 | 100 | 50 | 100 | 100 | 70 |

Weights Assigned to Exams

| Exam 1 | Exam 2 | Final |
|--------|--------|-------|
| 100 | 100 | 200 |

Note that quizzes are not listed above. Pop quizzes will take place when I see they are needed. Pop quizzes are always short exams testing if you have acquired the most basic knowledge of each topic. As a result, they usually take place about one week after a major topic is completely discussed. However, it does not mean every topic will have a quiz.

Grading

The theory part has two midterm exams and some quizzes. The sum of the scores you will obtain has a 50% weight. The programming part has five to six assignments, depending on our progress. The sum of the scores you will obtain constitutes the other 50%. Grading will be based approximately on the scale shown in the table below.

| Grade | A | AB | B | BC | C | CD | D | F |
|-------|---------|--------|--------|--------|--------|--------|--------|-------|
| Scale | 90-100% | 85-89% | 80-84% | 75-79% | 70-74% | 65-69% | 60-64% | 0-59% |

Website: 4/6

Exams and Final

You are responsible for avoiding exam time conflicts!!

ADA

MTU complies with all federal and state laws and regulations regarding discrimination, including the Americans with Disability Act of 1990 (ADA). If you have a disability and need a reasonable accommodation for equal access to education or services at MTU, please call Dr. Bonnie B. Gorman, Dean of Students, (2212). For other concerns about discrimination, you may contact your advisor, department head, or the Affirmative Action Office (3310).

The following two web sites may provide you with more information:

- [Disability Services](#)
- [Institutional Equality Office](#)

Other Useful Information

- [MTU Counseling Services](#)
- [Academic Calendar](#)

Downloadable Stuffs

Files are in HTML or PDF format.


- Programming Information
 - [Programming Assignment I](#)
 - [Programming Assignment II](#)
 - [Programming Assignment III](#)
 - [Programming Assignment IV](#)
 - [Programming Assignment V](#)
 - [Programming Assignment VI](#)
- Exam Information
 - [Exam 1 Solutions](#) (PDF file)
 - [Exam 2 Solutions](#) (PDF file)
- Reading Lists

Programming Assignments and EXAM Solutions

Weekly Reading Lists

| | | | | | | |
|------------------------|------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7 |
| Week 8 | Week 9 | Week 10 | Week 11 | Week 12 | Week 13 | Week 14 |

You are visitor **096527** since January 16, 2011
Last update: August 30, 2020

 Please send comments and suggestions to shene@mtu.edu

Website: 5/6

CS3331 Concurrent Computing Slides -- (Fall 2020 Version)

Always make sure you use the current version of slides for study as I update these slides EVERY semester. Moreover, if you cannot access a certain sets of slides, it means they are being updated and should be available soon.

| Number of Weeks | Course Topics | | Slides | No. of Slides | Total Slides |
|-----------------|---|---|---|---------------|--------------|
| 1 | Basics | | 01-Intro.pdf | 33 | 33 |
| 1 | Hardware and OS Stuff | | 02-Hardware-OS.pdf | 12 | 45 |
| 3 | Process and Threads | Process | 03-Process.pdf | 65 | 110 |
| | | Threads | 04-Thread.pdf | 23 | 133 |
| 1 | Critical Sections and Mutual Exclusion | | 05-Sync-Basics.pdf | 26 | 159 |
| 6 | Synchronization | Software and Hardware Solutions | 06-Sync-Soft-Hardware.pdf | 29 | 188 |
| | | A Very Brief C++ and ThreadMentor Review | 07-Some-Cpp-TM.pdf | 50 | 238 |
| | | Mutex Locks and Semaphores | 08-Semaphores.pdf | 78 | 316 |
| | | Race Condition: Revisited | 09-Race-Conditions.pdf | 34 | 350 |
| | | Monitors | 10-Monitors.pdf | 40 | 390 |
| | | Deadlock and Livelock | 11-Deadlock.pdf | 15 | 405 |
| 2 | Programming Languages and System Supports | Communication Channels | 12-Channels.pdf | 28 | 433 |
| | | Java Threads and Synchronization | 13-Java-Threads.pdf | 16 | 449 |
| | | Ada Tasking | 14-Ada-Tasking.pdf | 20 | 469 |
| | | Pthreads | 15-Pthreads.pdf | 17 | 486 |

Other Stuff

| Topics | Slides | Videos | No. of Slides | Total Slides |
|--------------------------------|---|--|---------------|--------------|
| Course Information | 3331-info.pdf | none | 12 | 12 |
| More Mutual Exclusion Examples | Mutual-Exclusion-Examples.pdf | CS3331-2020-1109-Mutual-Exclusion-Examples.mp4 | 23 | 35 |
| EXAM I Review | EXAM1.pdf | CS3331-2020-1110-EXAM1-Review.mp4 | 31 | 66 |
| EXAM II Review | EXAM2.pdf | CS3331-2020-1110-EXAM2-Review.mp4 | 32 | 98 |

Website: 6/6

Other Stuffs

| Topics | Slides | Videos | No. of Slides | Total Slides |
|--------------------------------|---|--|---------------|--------------|
| Course Information | 3331-info.pdf | none | 12 | 12 |
| More Mutual Exclusion Examples | Mutual-Exclusion-Examples.pdf | CS3331-2020-1109-Mutual-Exclusion-Examples.mp4 | 23 | 35 |
| EXAM I Review | EXAM1.pdf | CS3331-2020-1110-EXAM1-Review.mp4 | 31 | 66 |
| EXAM II Review | EXAM2.pdf | CS3331-2020-1110-EXAM2-Review.mp4 | 32 | 98 |

Videos

| Video Content | Date | Link |
|---------------------------------|--------------------|--|
| Basics | September 9, 2020 | CS3331-2020-0909 .mp4 |
| Hardware/OS and Process | September 14, 2020 | CS3331-2020-0914 .mp4 |
| Process and fork() | September 16, 2020 | CS3331-2020-0916 .mp4 |
| execvp() and Shared Memory | September 21, 2020 | CS3331-2020-0921-Process .mp4 |
| Shared Memory and Threads | September 23, 2020 | CS3331-2020-0923-Shared-Memory-Threads .mp4 |
| Threads, Part II | September 30, 2020 | CS3331-2020-1001-Multicore .mp4 |
| Mutual Exclusion | September 30, 2020 | CS3331-2020-0930-Mutual-Exclusion .mp4 |
| Software and Hardware Solutions | October 5, 2020 | CS3331-2020-1005-Software-Solutions .mp4 |
| Software and Hardware Solutions | October 7, 2020 | CS3331-2020-1007-Hardware-Solutions .mp4 |
| C++ and ThreadMentor Review | October 10, 2020 | CS3331-2020-1010-Cpp-Review .mp4 |
| C++ and ThreadMentor Review | October 14, 2020 | CS3331-2020-1013-TM-Review .mp4 |
| ThreadMentor Demo 1 | October 14, 2020 | CS3331-2020-1014-TM-Demo .mp4 |
| Semaphores: Concepts | October 16, 2020 | CS3331-2020-1016-Semaphores-Part-1-Concepts .mp4 |
| Semaphores: Examples | October 25, 2020 | CS3331-2020-1025-Semaphores-Part-2-Examples .mp4 |
| Semaphores: Baton Passing | October 31, 2020 | CS3331-2020-1031-Semaphores-Part-3-Pass-the-Baton .mp4 |
| Semaphores: ThreadMentor | October 31, 2020 | CS3331-2020-1031-Semaphores-Part-4-TM .mp4 |
| Semaphores: ThreadMentor Demo | November 2, 2020 | CS3331-2020-1102-Semaphores-Part-5-TM-Demo .mp4 |

Grade Distributions: 2011-2020

Worst Class and
Worst Evaluation Ever

| Grade | 11F | 12S | 12F | 13S | 13F | 14S | 14F | 15S | 15F | 16S | 16F | 17S | 17F | 18S | 18F | 19S | 19F | 20F |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|-------|
| A | 26 | 23 | 12 | 25 | 12 | 19 | 13 | 3 | 16 | 10 | 20 | 14 | 20 | 10 | 25 | 14.6 | 12.2 | 21.28 |
| AB | 20 | 9 | 4 | 33 | 19 | 21 | 15 | 3 | 6 | 10 | 22 | 7 | 16 | 24 | 4.2 | 9.8 | 9.8 | 17.02 |
| B | 11 | 20 | 16 | 17 | 14 | 12 | 9 | 5 | 22 | 12 | 5 | 10 | 14 | 16 | 10.4 | 12.2 | 19.5 | 8.51 |
| BC | 6 | 14 | 16 | 8 | 2 | 21 | 15 | 30 | 12 | 17 | 15 | 12 | 11 | 16 | 10.4 | 9.8 | 17.1 | 10.64 |
| C | 9 | 6 | 4 | 6 | 7 | 5 | 9 | 14 | 14 | 8 | 5 | 12 | 9 | 9 | 20.8 | 4.9 | 4.9 | 19.15 |
| CD | 3 | 9 | 8 | 3 | 7 | 5 | 9 | 11 | 6 | 12 | 11 | 12 | 9 | 7 | 2.1 | 17.1 | 4.9 | 10.64 |
| D | 6 | 6 | 24 | 0 | 24 | 7 | 4 | 14 | 10 | 21 | 2 | 19 | 5 | 14 | 18.8 | 14.6 | 22 | 8.51 |
| F | 20 | 14 | 16 | 8 | 14 | 10 | 28 | 22 | 12 | 12 | 20 | 14 | 16 | 5 | 8.3 | 17.1 | 9.8 | 4.26 |
| Size | 35 | 35 | 25 | 36 | 42 | 42 | 47 | 37 | 49 | 52 | 55 | 42 | 56 | 58 | 48 | 41 | 41 | 49 |

■ Data shown here only include students who completed this class.
Students who did not take the final exam were not included.

*It takes a really bad school to ruin a good student
and
a really fantastic school to rescue a bad student.*

Dennis J. Frailey

The End