General Information: 1/5

- **Course:** CS3331 Concurrent Computing
- **Office:** Rekhi 305
- **Instructor:** Ching-Kuang Shene aka *C-K*
- **Meeting:** M/W/F 2:05-2:55pm ***Chemical Science & Engineering 102***
- **Office Hrs:** M/T/W/Tr 3:00-3:30pm
- **Textbook:** No textbook; but, you will have my slides and web-based materials.
- **Exams:** *two* exams and *one* final
- **Programming:** *six* programming assignments
- **Others:** A few pop quizzes
- 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 or /local/cs3331.ck/common or /mtu/cs3331.ck/common will have all slides used in class, software, etc. Check it frequently.
- ***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 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

- We may switch to Canvas for program submission.
- 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 point, 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 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 quiz, 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.**
- **Leave you regrade requests in my mailbox in the Computer Science office.**
- **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.**

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 seven 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).**

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

Dennis J. Frailey

Grade Distributions: 2011-2018

Worst Class and Worst Evaluation Ever

Grade	11F	12S	12F	13S	13F	14S	14F	15S	15F	16S	16F	17S	17F	18S	18F
A	26	23	12	25	12	19	13	3	16	10	20	14	20	10	25
AB	20	9	4	33	19	21	15	3	6	10	22	7	16	24	4.2
B	11	20	16	17	14	12	9	5	22	12	5	10	14	16	10.4
BC	6	14	16	8	2	21	15	30	12	17	15	12	11	16	10.4
C	9	6	4	6	7	5	9	14	14	8	5	12	9	9	20.8
CD	3	9	8	3	7	5	9	11	6	12	11	12	9	7	2.1
D	6	6	24	0	24	7	4	14	10	21	2	19	5	14	18.8
F	20	14	16	8	14	10	28	22	12	12	20	14	16	5	8.3
Size	35	35	25	36	42	42	47	37	49	52	55	42	56	58	48

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