

# CS3211 – Parallel and Concurrent Programming

Admin Information

# Teaching Staff – **Lecturer**

- Dr. Cristina Carbunaru
- Contact Information:

Phone number: 65168850

Email: **ccris@comp.nus.edu.sg**

- Comments / suggestions welcome
- Email in advance to book a consultation slot

# Teaching Assistants

- Sriram Sami
- Walter Kong
- Ng Zhia Yang
- Seow Alex
- Mayank Keoliya
- Kingsley Kuan Jun Hao
- Teh Xue Yong
- Loh Jing Yen
- Travis Toh Guan Wei
- Audrey Felicio Anwar

# Overview

- CS3211 teaches the challenging field of parallel and concurrent programming
- The world is concurrent! (Joe Armstrong)
- At the end of the module
  - Understand and confidently use concurrency in major programming paradigms popular nowadays
  - Write programs understanding the concurrency overhead
  - Understand safety constraints, robustness, and
  - Know the differences among different concurrency paradigms.

# Not just one topic...

- Cover new parallel and concurrent programming paradigms and programming languages
  - C++20
  - Go Lang
  - Rust
- Full set of principles and challenges that need to be followed and addressed in any piece of concurrent software.

Concurrent programming is pervasive nowadays  
and software development should be carefully crafted with this in mind.

# Concurrency is pervasive

- Not only in the real-world
- Our computers have
  - Several cores + several types of memories
  - Operating system
  - Compilers
  - Programming languages
- Concurrent code often runs in parallel

# Learning Outcomes

By the end of the module, the students should be able to

- explain the concurrent programming challenges.
- define and correctly use different synchronization mechanisms.
- apply concurrent programming principles in programs in different programming languages.
- define and identify different issues in the concurrent programs.
- adapt the concurrent programming principles to new programming languages or paradigms.

# Assessment

- 8% - Tutorial attendance
- 12% - 4 Quizzes
- 40% - 3 Assignments
  - Modern C++ (C++20) ~20%
  - Golang ~ 10%
  - Rust ~ 10%
- 40% - In-person open-book final exam



Week	Lecture (Tue, 12-2pm, LT19)	Tutorial / Lab (Wed-Fri, COM1-B1-02)	Deadlines
1	Introduction to Concurrency		
2	Threads, Tasks, and Synchronization Primitives in Modern C++	T0.5: Startup with Modern C++ (no class)	
3	PH: CNY (no class)	T1: Threads and Synchronization in C++ Release A1	
4	Shared Memory and Lock-free Programming in Modern C++	T2: Atomics in C++	Sun, 8pm - Quiz 1
5	Examples of Lock-based and Lock-free Programming	T3: Lock-free Programming in C++	
6	Testing and Debugging Concurrent C++ Programs	T4: Debugging	Sun, 8pm - Quiz 2
Recess		T4.5: Startup with Go (no class)	Sat, 2pm – Assignment 1
7	Concurrency in Go	T5: Introduction to Go Release A2	
8	Concurrency Patterns in Go	T6: Synchronization in Go	
9	Classic Synchronization Problems in C++ and Go	T7: Comparison of Different Synchro Problems in C++ / Go	Sun, 8pm - Quiz 3
10	Safety in Rust	T7.5: Startup with Rust (no class) T8: Introduction to Rust	Sat, 2pm – Assignment 2
11	Asynchronous Programming in Rust	T9: Asynchronous Programming in Rust Release A3	Sun, 8pm - Quiz 4
12	Formal Verification and Model Checking	PH: Good Friday (no class)	
13	Recap	T10: Implementing Synchronization Primitives	Sat, 2pm - Assignment 3
Exam	In-person Open-book Exam	CS3211 L0 - Admin	4 May, 9am - Exam 9

# References

- C++ Concurrency in Action - Practical Multithreading, Second Edition
  - Author: Anthony Williams
  - First 6 weeks of the semester

# Minimal overlap with other modules

- CS3210
- CS4231
- ?

# Why study this module?

- Industry writes concurrent code in many of the programming languages that we cover
- Tour of different languages for concurrency
  - Look at different PLs and how concurrency is achieved in these PLs
- Make your code have better performance
  - There are many aspects to performance