



Prescription

This course examines a range of techniques for programming multi-threaded and distributed applications. Topics include synchronisation mechanisms used for programs that communicate via shared memory and message passing techniques for programs that communicate across a network. Practical work involves implementing programs using these techniques in a modern concurrent language, such as Java.

Course learning objectives

Students who pass this course will be able to:

1. Identify the different types of concurrency problems that can occur in both shared memory models and message passing models. (3(f))
2. Analyse a given program to identify where and why these problems might arise. 3(c))
3. Apply key concurrency control mechanisms and algorithms to avoid these problems in a concurrent program. (3(b))
4. Implement solutions to concurrent problems in code using current programming languages such as Java. (3(b)), 3(f))
5. Evaluate the "fitness to purpose" of solutions to concurrency problems using reasoning from first principles and experimentation. (3(b), 3(d))
6. Communicate the rationale for their design, implementation choices and the meaning of the results of their testing in project reports. (2(b))

Course content

The course is primarily offered in-person, but there will also be a remote option and there will be online alternatives for all the components of the course for students who cannot attend in-person.

Students taking this course remotely must have access to a computer with camera and microphone and a reliable high speed internet connection that will support real-time video plus audio connections and screen sharing. Students must be able to use Zoom; other communication applications may also be used. A mobile phone connection only is not considered sufficient. The computer must be adequate to support the programming required by the course: almost any modern windows, macintosh, or unix laptop or desktop computer will be sufficient, but an Android or IOS tablet will not.

If the assessment of the course includes tests, the tests will generally be run in-person on the Kelburn campus. There will be a remote option for students who cannot attend in-person, but the remote option imposes extra costs on the School and will be limited to students with a strong justification (for example, being enrolled from overseas). The remote test option will use the ProctorU system for online supervision of the tests. ProctorU requires installation of monitoring software on your computer which also uses your camera and microphone, and monitors your test-taking in real-time. Students who will need to use the remote test option must contact the course coordinator in the first two weeks to get permission and make arrangements.

=====

A summary of the topics is:

1. Threads, thread pools and tasks
2. Semaphores and monitors
3. Futures, fork-join, thread pools and tasks.
4. Dead lock and starvation
5. Critical sections and the difficulty of implementing them correctly
6. Applying concurrency concepts to develop concurrent programs
7. Differences between concurrent, distributed and parallel programs
8. Shared memory parallelism
9. Message passing systems

Required Academic Background

A good grasp of the Java programming language is required to participate in this course.

This includes understanding the following concepts:

- divide logic in methods;
- test code;
- handling (and releasing!) resources;
- functional understanding of methods as input/output reusable black boxes;
- divide et impera algorithms, like sorting a list using merge sort;
- difference between in place computation and out of place computation;
- basic generics: no wildcards or intricate generic types, but you need to be comfortable defining your own generic classes;
- Lists/Sets/Maps need to be a very obvious tool;
- confidence with Java lambdas and the general idea that an object can be just a unit of behaviour;
- capacity to read documentation in the form of official Javadocs;
- exceptions, difference between checked and unchecked exceptions; most common programming patterns about exceptions.

Withdrawal from Course

Withdrawal dates and process:

<https://www.wgtn.ac.nz/students/study/course-additions-withdrawals>

Lecturers

Marco Servetto (Coordinator)

marco.servetto@vuw.ac.nz 04 4635820

258 Cotton, Kelburn

Qiang Fu

qiang.fu@vuw.ac.nz 04 4635233 ext 8829

414 Alan MacDiarmid Building, Kelburn

Teaching Format

This course will be offered in-person and online. For students in Wellington, there will be a combination of in-person components and web/internet based resources. It will also be possible to take the course

entirely online for those who cannot attend on campus, with all the components provided in-person also made available online.

During the trimester there will be three lectures per week.
Some weeks the Thursday lecture will be replaced by a tutorial/lab.

Student feedback

Student feedback on University courses may be found at:
www.cad.vuw.ac.nz/feedback/feedback_display.php

Dates (trimester, teaching & break dates)

- Teaching: 22 February 2021 - 28 May 2021
- Break: 05 April 2021 - 18 April 2021
- Study period: 31 May 2021 - 03 June 2021
- Exam period: 04 June 2021 - 19 June 2021

Class Times and Room Numbers

22 February 2021 - 04 April 2021

- **Tuesday** 13:10 - 14:00 – LT103, Hugh Mackenzie, Kelburn
- **Thursday** 13:10 - 14:00 – LT103, Hugh Mackenzie, Kelburn
- **Friday** 13:10 - 14:00 – LT220, Hunter, Kelburn

19 April 2021 - 30 May 2021

- **Tuesday** 13:10 - 14:00 – LT103, Hugh Mackenzie, Kelburn
- **Thursday** 13:10 - 14:00 – LT103, Hugh Mackenzie, Kelburn

Other Classes

According to the Lecture Schedule, some week instead of the Lecture on Thursday, there will be a guided lab.

Set Texts and Recommended Readings

Required

There are no required texts for this offering.

Recommended

There is no official course textbook.

Mandatory Course Requirements

In addition to achieving an overall pass mark of at least 50%, students must:

- achieve across the assignments an average of at least 40%.

- achieve at least a 'D' grade in the two term tests.

If you believe that exceptional circumstances may prevent you from meeting the mandatory course requirements, contact the Course Coordinator for advice as soon as possible.

Assessment

There will be four assignments and two term tests.

Assignments

Each assignment will consist of written questions and may include some practical programming problems that explore different aspects of the material presented in the lectures. Doing the assignments is an crucial part of the learning process and is essential for a proper understanding of the material.

Term Tests

The term tests will evaluate your capacity to apply the concepts of the course in a controlled environment with precise time limits. Paper non-English to English dictionaries will be permitted.

Assessment Item	Due Date or Test Date	CLO(s)	Percentage
Assignment 1(Sort)	Week 4	CLO: 1,2,3,4,5,6	8%
Assignment 2(Universe)	Week 7	CLO: 1,2,3,4,5,6	15%
Term Test 1	Week 8	CLO: 1,2,3,4,5,6	24%
Assignment 3(Quiz)	Week 9	CLO: 1,2,3,4,5,6	8%
Assignment 4(Actors)	Week 11	CLO: 1,2,3,4,5,6	15%
Term Test 2(2 hours duration)	TBC	CLO: 1,2,3,4,5,6	30%

Penalties

Late submissions will be penalised 20% of their mark for every day overdue. This means after 5 days zero marks will be awarded. In this case, the work should still be submitted in order to pass the mandatory requirements. However, submissions will not be accepted once any model answers have been given out.

Extensions

Individual extensions will only be granted in exceptional personal circumstances, and should be negotiated with the course coordinator before the deadline whenever possible. Documentation (eg, medical certificate) may be required.

Submission & Return

All work should be submitted through the ECS submission system, accessible through the course web pages. Marks and comments will be returned through the ECS marking system, also available through the course web pages.

Workload

In order to maintain satisfactory progress in NWEN 303, you should plan to spend an average of 10 hours per week on this paper. A plausible and approximate breakdown for these hours would be:

- Lectures and tutorials/labs: 3 hours
- Readings: 1-2 hours
- Assignments: 6-7 hours

Teaching Plan

See https://ecs.wgtn.ac.nz/Courses/NWEN303_2021T1/LectureSchedule

Communication of Additional Information

All online material for this course can be accessed at https://ecs.wgtn.ac.nz/Courses/NWEN303_2021T1/

Links to General Course Information

- Academic Integrity and Plagiarism: <https://www.wgtn.ac.nz/students/study/exams/integrity-plagiarism>
- Academic Progress: <https://www.wgtn.ac.nz/students/study/progress/academic-progress> (including restrictions and non-engagement)
- Dates and deadlines: <https://www.wgtn.ac.nz/students/study/dates>
- Grades: <https://www.wgtn.ac.nz/students/study/progress/grades>
- Special passes: Refer to the Assessment Handbook, at <https://www.wgtn.ac.nz/documents/policy/staff-policy/assessment-handbook.pdf>
- Statutes and policies, e.g. Student Conduct Statute: <https://www.wgtn.ac.nz/about/governance/strategy>
- Student support: <https://www.wgtn.ac.nz/students/support>
- Students with disabilities: https://www.wgtn.ac.nz/st_services/disability/
- Student Charter: <https://www.wgtn.ac.nz/learning-teaching/learning-partnerships/student-charter>
- Terms and Conditions: <https://www.wgtn.ac.nz/study/apply-enrol/terms-conditions/student-contract>
- Turnitin: <http://www.cad.vuw.ac.nz/wiki/index.php/Turnitin>
- University structure: <https://www.wgtn.ac.nz/about/governance/structure>
- VUWSA: <http://www.vuwsa.org.nz>

Offering CRN: [17182](#)

Points: 15

Prerequisites: ENGR 123 or MATH 161; SWEN 221; 15 points from (COMP 261, CYBR 271, NWEN 241, 243)

Duration: 22 February 2021 - 20 June 2021

Starts: Trimester 1

Campus: Kelburn