



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

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

Withdrawal from Course

Withdrawal dates and process:

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

Lecturers

Marco Servetto (Coordinator)

marco.servetto@vuw.ac.nz 04 4635820

258 Cotton, Kelburn

Teaching Format

During the trimester there will be two lectures and one tutorial/lab per week.

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: 04 March 2019 - 09 June 2019
- Break: 15 April 2019 - 28 April 2019
- Study period: 10 June 2019 - 13 June 2019
- Exam period: 14 June 2019 - 29 June 2019

Class Times and Room Numbers

04 March 2019 - 14 April 2019

- **Monday** 11:00 - 11:50 – LT103, Hugh Mackenzie, Kelburn
- **Tuesday** 11:00 - 11:50 – LT103, Hugh Mackenzie, Kelburn
- **Thursday** 11:00 - 11:50 – LT103, Hugh Mackenzie, Kelburn

29 April 2019 - 09 June 2019

- **Monday** 11:00 - 11:50 – LT103, Hugh Mackenzie, Kelburn
- **Tuesday** 11:00 - 11:50 – LT103, Hugh Mackenzie, Kelburn
- **Thursday** 11:00 - 11:50 – LT103, Hugh Mackenzie, Kelburn

Other Classes

Every week there will be either a tutorial or a lab.

Set Texts and Recommended Readings

Required

There are no required texts for this offering.

There is no official course textbook. However, in previous years, we have used the textbook "*Principles of Concurrent and Distributed Programming*" (Second edition) by M. Ben-Ari and this may be useful for a deeper view on some of the topics covered in this course. The book is available online through the VUW Safari book subscription.

Mandatory Course Requirements

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

- achieve across the two projects an average of at least 40%.
- achieve at least a 'D' grade in the final exam.

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 three assignments, two projects and a final examination.

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. Marking will be based upon the correctness of your answers against a marking guide.

Projects

Each project will require you to design, implement and evaluate a system that solves a given problem. We will set the goals for the project and assist with methodology without directly specifying how work should be completed. A crucial part of the project is the writing up of what you have done in a laboratory report that will be submitted as well as the program code and test results. The first project will focus on concurrent programming and the second project will focus on parallel programming.

The projects will be marked using a marking guide that will focus on the achievement of goals, the application of methodologies appropriate to the course and the problem, and your evaluation of what you have produced. The report will also be used to evaluate your ability to write in a professional manner.

Final Examination

The final examination will be two hours long. No computers, electronic calculators or similar device will be allowed in the final examination. Paper non-English to English dictionaries will be permitted.

Assessment Item	Due Date or Test Date	CLO(s)	Percentage
Three assignments		CLO: 1,2,3,4,5,6	12%
Two projects		CLO: 1,2,3,4,5,6	28%
Final examination		CLO: 1,2,3,4,5,6	60%

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
- Assignment and project work: 6-7 hours

Teaching Plan

See https://ecs.victoria.ac.nz/Courses/NWEN303_2019T1/LectureSchedule

Communication of Additional Information

All online material for this course can be accessed at https://ecs.victoria.ac.nz/Courses/NWEN303_2019T1/

Links to General Course Information

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

Offering CRN: [17182](#)

Points: 15

Prerequisites: ENGR 123 or MATH 161: 30 points from (COMP 261. NWEN 241. 243. SWEN 221)

Duration: 04 March 2019 - 30 June 2019
Starts: Trimester 1
Campus: Kelburn