NWEN 241
Systems Programming

2024 Trimester 1

Jyoti Sahni

Jyoti.sahni@ecs.vuw.ac.nz

Contents

Introduction to the course:

- Who are the people?
- What’s the course about?
- How does the course work?

People (1)

Course Coordinator

Jyoti Sahni
jyoti.sahni@ecs.vuw.ac.nz
AM414 / https://vuw.zoom.us/my/jyotisahni

NWEN 241 Office Hours: Tue, 10:00-12:00

Lecturer

Alvin Valera
alvin.valera@ecs.vuw.ac.nz
AM418 / https://vuw.zoom.us/my/alvin.valera

NWEN 241 Office Hours: Tue, 10:00-12:00

People (2)

Tutors

- Arnav Dogra
- Buddhima Amarathunga
- Christopher Jones
- Kahurangi Burkitt
- Nicholas Winsley
- Nicola Kerswill
- Patrick Mills
- Rene Sharma
- Richard Whitmarsh
- Vinh Nguyen

What’s the course about?

Programming in "low/mid level" languages – C and C++

- “True” low-level programming is either machine code (which the CPU directly understands), or assembly code (which is a simple one-to-one text translation of the binary code that the CPU understands).
- High level languages have a more natural and readable syntax –Each instruction is generally translated into multiple lines of machine code to be executed by the computer’s processor.
- With high level languages many tasks are taken care for us by the language libraries – e.g. type checking, memory management (such as garbage collection).
- With low level languages we need to know we have to manage our own application memory, it is not managed for us by the language.

What’s the course about?

Programming in "low/mid level" languages – C and C++

- What’s the advantage of “low/mid” level programming?
  - Efficiency and power consumption.
- C and C++ are often referred to as low-level languages due to the level of control they offer over system resources like memory: direct memory manipulation and management.
- However, they are also considered middle-level languages because they provide a balance of high-level abstraction and low-level control, portability and standard libraries for common tasks such as I/O, string manipulation and mathematical computations.
- With low level languages we need to know we have to manage our own application memory, it is not managed for us by the language.

What’s the course about?

3 broad sections:

1. C Programming concepts

2. A brief overview of System calls and Socket programming in C

3. C++ Programming concepts

Course Learning Objectives

- Use appropriate tools for compiling/debugging C/C++ programs.
- Write C programs using pointers and arrays, user-defined data types, input/output operations, bit-level operations, and user-defined and library routines.
- Understand the differences between C and C++, and write C++ programs using stream input/output, classes, vectors and templates.
- Use or understand the main techniques of dynamic memory management in C and C++.
- Structure larger programs in multiple files.
- Understand the differences between application software and system software.

Course Assumptions

- This course assumes that you are familiar or have taken courses that have dealt with the following topics:
  - Binary representation of numbers
  - Basic logic or Boolean algebra
  - Computer program design
  - Java programming

- Computer program design and Java programming are essentially covered in COMP 102 and COMP 103

- If you want to brush up on your knowledge of binary representation and basic logic: https://www.bottomupcs.com/chapter01.xhtml

Course Delivery

- **2 Lectures**
  - Tuesdays and Wednesdays
  - 9:00-9:50 at HMLT205
  - Recorded and accessed via Nuku
- **1 Tutorial-Style Lecture**
  - Fridays
  - 9:00-9:50 at HMLT205
  - Recorded and accessed via Nuku
- **Helpdesk (from Weeks 2-12)**
  - Helpdesk @CO246 (in-person)
  - If you a remote student, write to me
  - First-come, first-served

See Course Wiki for more details about Lecture Schedule

See Course Wiki for more details about Helpdesk Schedule

In-Person Helpdesk Sessions

13 18
14 17
15 16
19 24
20 23
21 22
25 30
26 29
27 28
10 9
11 8
12 7
4 3
5 2
6 1

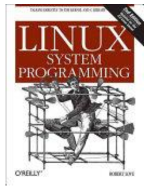
- Lab: Systems and Network Lab (CO246)
  - ID access cards (Swipe Cards)
  - Should work if you are registered in NWEN 241
- PC workstations
  - Linux operating system, KDE as graphical user interface
  - Network file system: you can access your files from any of the PCs
  - Compilers & debuggers: gcc, g++, gdb, and more
  - Text editors: kate, gedit, emacs, vi, vim, and more
- Text editor vs IDE: Text editor
- Remote access: https://ecs.victoria.ac.nz/Support/TechNoteWorkingFromHome

Books and Other Resources (1/2)

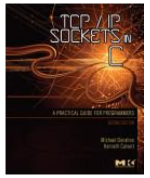
- No textbook required
- Good references:

Perry, Gregory, **C Programming Absolute Beginner's Guide**, Third Edition
Available Online

Kochan, Stephen, **Programming in C,** Fourth Edition
Available Online



Love, Robert, **Linux System Programming,** 2nd Edition
Available Online



Donahoo, Michael,
TCP/IP Sockets in C, 2nd Edition
Available Online

Malik D.S., **C++ Programming: Program Design Including Data Structures**
Available Online

Books and Other Resources (2/2)

- Classic C programming reference:

Kernighan and Ritchie, **The C Programming Language**, 2nd Edition
Hardcopy Available in VUW Library

- Lecture slides:
  - To be released a day before lecture in course website

Assessment

Component	Weight (%)
4 Exercises (2.5% each)	10
4 Assignments (5% each)	20
Mid-Term Test (Week 6)	15
Final Exam (assessment period)	55

Mandatory Course Requirements

- Submit a reasonable attempt at 3/4 of the assignments.
- Obtain a **D** grade or better in the final test.

Exercises

- Simple guided programming tasks
- *Automatically marked* when you submit them to the online submission system
- You may re-submit as many times as you like in order to improve your mark before the deadline
- Exercise submitted after the deadline will not be marked and will get 0 marks.
    - See course wiki for handout dates and deadlines (Generally, due on Wednesdays at 23:59)

Assignments (1)

- **Programming tasks** to test your practical knowledge

- 4 assignments in total, each assignment weighing 5% of the final grade
  - See course wiki for handout dates and deadlines
  - Due on Mondays at 23:59 (except Assignment 4, due on Sunday 23:59)
  - Marking (in-person) will start the following Monday (except for Assignment 4, which starts the following day)
  - You must get your assignment marked at a marking session

Assignments (2)

“Late days” credit

- Each student will have 3 “late days” which you may choose to use for any lab assignment(s) during the course. No “late days” for exercises
- There will be no penalty applied for these late days
- You do not need to apply for these - any late days you have left will be automatically applied to lab assignments that you submit late
- The late days are intended to cover minor illnesses or other personal reasons for being late. You should only ask for extensions in the case of more significant or longer lasting problems (and you may need documentation)
- Do not waste “late days” on procrastination!

Assignments (3)

Penalties for late submission (beyond the late days):

- Each late submission will be penalised by
  - 20% of the achieved marks if it is up to 24 hours late, and
  - 40% if it is between 24 hours and 48 hours late.
- Any work submitted more than 48 hours after the deadline will receive 0 marks
- Unless an extension is granted. Extension requests must be supported by valid and reasonable justifications.
  - Extension requests can be made through the ECS submission system

Plagiarism (1)

You are encouraged to discuss the principles of the course and assignments with your tutors and other students, to help and seek help with programming details, problems involving the lab machines etc.

- **However, any work you hand in must be your own work**
- Plagiarism is **NOT tolerated** (Read School policy on Plagiarism)
  - Presenting someone else's work as your own, including
    - material from a published source such as a library book, a journal article, etc.
    - material from an on-line software library, web pages, etc.
    - the work of another student, friend, relative, etc.

Plagiarism (2)

Penalties:

- zero marks for the work to which it relates
- greater penalties in accordance with the University's Statute on Student Conduct
- Students who knowingly allow other students to copy their work may also be penalised

If you got help or code from other students, non-students, web-sources, etc. state it clearly to avoid plagiarism issue.

Plagiarism (3)

AI Red - Do not to use AI tools (ChatGPT, Bing Chat, Github Copilot, Google Bard, Moonbeam, etc...) to generate submitted material, or complete coursework in this course.

- Relying on AI tools may hinder your understanding of the subject matter.
- If you choose to use them, it is imperative that you fully comprehend the functionality of your code. Assignments will be evaluated in person, requiring you to articulate and explain your approach.
- Tell us if you used them. Failure to disclose the use of AI tools and subsequent discovery of their use will result in penalties.

Getting Help for Assignments & Exercises

1. For quick questions, flick an e-mail to nwen241-staff@ecs.vuw.ac.nz

2. For issues that are harder to resolve, attend any Helpdesk session (starting on Week 2, first-come first-served)

Do not wait until the last Helpdesk!

Mid-Term Test and Final Exam

- Mid-Term Test and Final Exam will be in-person
- See course wiki for the test dates
- Venue to be announced closer to the test dates

Course Wiki

- Link: https://ecs.victoria.ac.nz/Courses/NWEN241_2024T1/WebHome
- **Check the course wiki regularly!**

APPOINT A CLASS REP

Become a Class Representative at Victoria University of Wellington!

Class Reps are the bridge between the course coordinator, lecturer, and the class to support and improve students' learning experiences in your course and at Victoria.

REGISTER online: http://www.vuwsa.org.nz/class-representatives/

Representing your class has many benefits; VicPlus points, Class Rep certificates, professional and personal growth, and links to other representation opportunities.

Class Representative(s)

Please e-mail me if you want to be a class representative