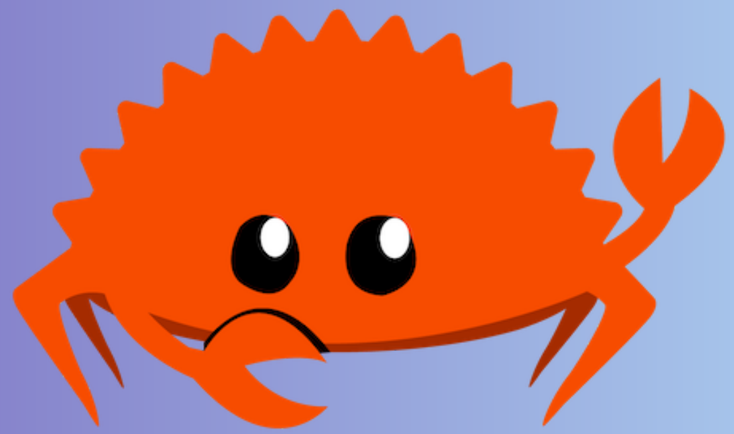
COMP6991 24T1

Course Introduction



Thematic Quotes

A PROGRAMMING LANGUAGE IS A TOOL THAT HAS PROFOUND INFLUENCE ON OUR THINKING HABITS.

(Dijkstra’s algorithm, foundations of concurrency)

Edsger W. Dijkstra

A POWERFUL PROGRAMMING LANGUAGE IS MORE THAN JUST A MEANS FOR INSTRUCTING A COMPUTER TO PERFORM TASKS.

THE LANGUAGE ALSO SERVES AS A FRAMEWORK WITHIN WHICH WE ORGANIZE OUR IDEAS ABOUT PROCESSES.

(Creative commons, Free Software Foundation)

Hal Abelson

COMPUTER LANGUAGE DESIGN IS JUST LIKE A STROLL IN THE PARK… JURASSIC PARK, THAT IS.

(Perl programming language, patch)

Larry Wall

COMP6991 Staff

CONVENOR

Dr. Andrew Taylor

LECTURER

Zac Kologlu

ADMINISTRATION

Shrey Somaiya, Tom Kunc, ...

TEACHERS

Alex Miao, Brian Li, Daniel Field, Daniel Wang, Ethan Dickson, Giulia Betke, Hanyuan Li, Jaden Lanceman, James Appleton, Jared Lohtaja, Jayden Leung, Jennifer King, Larry Tang, Luke Fisk-Lennon, Matthew Kokolich, Patrick Hao, Peter Derias, Gabriel Xu, Sabine Lim, Shrey Somaiya, Tom Kunc, Wisesa Resosudarmo, Xavier Carey, Zachary Ecob

Lots of programming!

Hopefully multiple languages

Hopefully different styles

Programming languages as a tool.

Language design, language choice.

What works well for you, and when?

Reason about language tooling & design.

Understand and work with program contracts.

Reason about invariants and correctness statically.

Have an additional language under your belt (Rust!)

... AFTER COMP6991

Course themes

PROGRAM DESIGN

PROGRAMMING LANGUAGES

PROGRAM SAFETY

PROGRAM ROBUSTNESS

Course Reading

NO STRICT REQUIREMENT

However some activities may include suggested reading (e.g. articles, blog posts)

RECOMMENDED READING

The Rust Programming Language, by Steve Klabnik and Carol Nichols, with contributions from the Rust Community

SUGGESTED READING

Rust in Action: Systems programming concepts and techniques, by Tim McNamara

Rust for Rustaceans: Idiomatic Programming for Experienced Developers, by Jon Gjengset; complex reading for more intermediate Rust programmers

Systems and Tools

RUST TOOLCHAIN

Rustup recommended if installing at home.
Cargo as a package manager (`6991 cargo` on CSE).

CRATES.IO

For third-party dependencies

RUSTDOC

e.g. https://doc.rust-lang.org/stable/std/ for standard library

AUTOTEST

`6991 autotest` on CSE

GIVE-CRATE

`6991 give-crate` on CSE

Lectures

TIMING

Monday & Tuesday, 18:00-20:00

LOCATION

Colombo Theatre A (K-B16-LG03)

RECORDING

Echo360 via Moodle (link on course website)
22T3 lectures as a backup.

SLIDES

On the course website

QUESTIONS

You can always interrupt with questions / comments!
(I love winding discussions, please do jump in!)

Lectures

FOCUS

On the thinking behind Rust's design and features.

LANGUAGES

Well, Rust -- obviously!
I'd like to examine many other languages too.
Be prepared to be guided through unfamiliar code!

FORMAT

Lots of babbling about.
Live coding examples will be common.
I will not just be teaching you to write Rust!
Self-directed learning is expected.

Workshops

WHEN / WHERE

2 hour workshops (listed as LAB on myUNSW).
Wednesdays through Fridays, weeks 1-5,7-10.
Almost all workshops are **on-campus**!
There is one online workshop, hosted on Discord
Invite URL is on the timetable page

FAQ

Workshops are not mandatory.
Workshops do not count towards any marks.
Workshops are not to work on assessable content.

... and yet, you should **absolutely** go!

Workshops

CONTENT

Very practical sessions.
Two course teachers there to help!
Focusing on slightly larger problems in small groups.

RETRO

Workshops often finish with a (surprisingly fun) retrospective.
What went well? What was a nightmare!?
Would this look any different in another language?

EXPECTATIONS

To have watched the current week's lectures.
To have a willingness to work on workshop material (not your weekly exercises, assignments).
Instead, direct those questions to the course forum!

Weekly Exercises

WHEN / WHERE

Weeks 1-5,7-9 (8 total), on the course website

CONTENT

Usually several exercises, mostly small programs.
Intended to help you practice **writing** Rust yourself.
Most questions are automatically marked.
Sometimes theoretical / open-ended questions, which are manually marked.

TESTING / SUBMISSION

Usually `6991 autotest` is available.
Usually submitted with `6991 give-crate`.

Blog Posts

ABOUT

Ongoing and entirely optional.
Help recoup up to 50% of lost marks from weekly exercises.
Posted on the course forum.
Read more on the course website.

Assignments

ASSIGNMENT 1

Program design.
Released around Monday week 4.
Due Monday week 7.
Worth 20% of your final grade.

ASSIGNMENT 2

Concurrent programming.
Released around Monday week 7.
Due Friday week 10.
Worth 25% of your final grade.

ADMINISTRIVIA

Individual assignments (no group work).
Standard UNSW late penalty applies, calculated hourly (in your advantage).

Final Exam

FORMAT

The exam will be online (assuming the school doesn't change its mind) and "open-book".
A mix of practical / theoretical questions.
Expect open-ended theory; often no "right answer", but answers must be well justified.

PAST PAPERS

We should be able to release 4 "past papers".
Past theory:prac ratio == **5:2**.
22T3, 23T1, 23T3 exams will be similar but not necessarily exact.

SCHEDULING

Expected to be 3 hours.
Centrally timetabled.

Code of Conduct

CSE offers an inclusive learning environment for all students.

In anything connected to UNSW, including social media, these things are student misconduct and will not be tolerated:

- racist/sexist/offensive language or images
- sexually inappropriate behaviour
- bullying, harassing or aggressive behaviour
- invasion of privacy

Show respect to your fellow students and the course staff

Plagiarism

MISCONDUCT

Cheating of any kind constitutes academic misconduct and carries a range of penalties.

EXAMPLES

Groupwork on individual assignments (discussion OK).
Allowing another student to copy your work.
Getting your hacker cousin to code for you.
Purchasing a solution to the assignment.

Plagiarism

OWN WORK

Weekly exercises, assignments, exam must be entirely your own work.
You can not work on assignments as a pair or group.

PENALTY

Plagiarism will be checked for and penalized.
Plagiarism may result in suspension from UNSW.
Scholarship students may lose scholarship.
International students may lose visa.
Supplying your work to any another person may result in loss of all your marks for the lab/assignment.

Assessment

WEEKLY EXERCISES
20%

+

ASSIGNMENT 1
20%

+

ASSIGNMENT 2
25%

+

FINAL EXAM
35%

} =

YOUR COMP6991 FINAL MARK

To pass: 50/100 overall
No hurdle requirements

Get the most out of COMP6991

LECTURES

Attend the lectures, and be involved!
Ask lots of questions, think about questions posed.

HAVE AN OPEN MIND

COMP6991 does not exist to convince you Rust is the One True Programming Language.
We want you to **think about** how you program; what works best for you!
We will explore many of Rust's weaknesses along the way.

If you leave COMP6991 hating Rust, but having become a better programmer along the way, we are happy.

MISC

Attend and actively participate in workshops.
Complete all weekly exercises.
Work hard on your assessments!

Assumed Knowledge

NON-NEGOTIABLE

Knowledge of C (COMP2521, COMP9024).
Worked on a large programming project (assignment).
Willingness to learn, open-mindedness.

HELPFUL TO HAVE

Experience programming in language(s) other than C.
Pain in programming!
Experience at different programming "levels"
(e.g. low-level, high-level).

Previous students say...

DIFFICULT COURSE

This course tackles difficult concepts, and has a relatively high workload.
Having experience with various programming languages **definitely** helps.
Often best taken towards the end of your degree.

BUT WITH THAT...

"thanks for the course, what a great note to end my uni degree on ! :D"

"Very happy with the course, the attitude of teaching staff, great learning community. Truly enjoyed it."

"The course structure was fantastic. The workshops are a truly innovative idea when it comes to teaching. Assessment tasks were great."

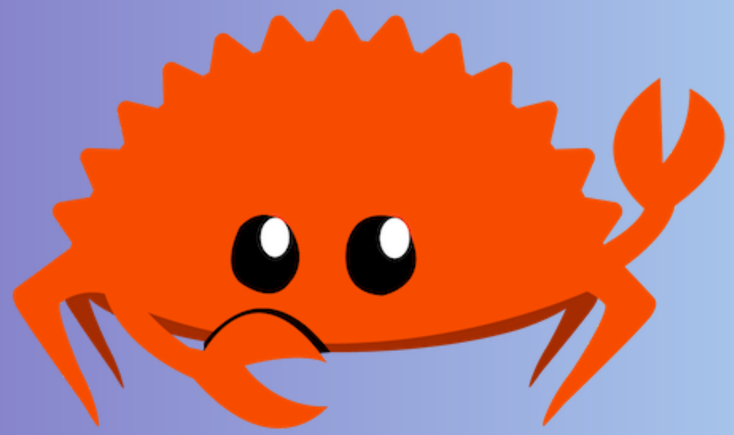
"very pog"

Course tour

COURSE WEBSITE

It all starts here, let's have a look!

https://cgi.cse.unsw.edu.au/~cs6991/24T1/



Let's have a great 24T1!