

Cornelia P. Inggs W.H.K. Bester

Programme & Project Introduction

RW797/771: HONOURS IN COMPUTER SCIENCE

Monday, 7 February 2022



Stellenbosch

UNIVERSITY
IYUNIVESITHI
UNIVERSITEIT

Postgraduate Coordinator

Cornelia Inggs

A509, General Engineering

cinggs@cs.sun.ac.za

Honours Project Coordinator

Willem Bester

A517, General Engineering

whkbester@cs.sun.ac.za

Questions

- ▶ E-mail us, send a Teams message, or schedule a meeting
- ▶ Use the SunLearn forum
- ▶ Speak to the Class Representative

The Honours in Computer Science programme consists of 128 credits.

1. You **MUST** pass 6 semester subjects ($6 \times 16 = 96$ credits)
 - ▶ At least 4 subjects from CS
 - ▶ More about the subject choices later
2. You **MUST** pass the year project (32 credits)
 - ▶ This is the killer!
 - ▶ The project descriptions are on the web
3. You **MUST** be a student assistant for at least one undergraduate CS module
4. You **SHOULD** have fun!

List of modules: cs.sun.ac.za/teaching/honours

List of projects: cs.sun.ac.za/courses/cs771

Deadline: 10 February

- ▶ Complete the module selection form: forms.gle/sVA7U2XAxWtUcXz46
(it's needed for registration; you can still change later)

Deadline: 11 February

- ▶ Complete the project selection form: forms.gle/MvtBtonGyW57MW/r8A
- ▶ Find an undergraduate course to dem

Deadline: 14 February

- ▶ Elect a class representative

Deadline: 25 February

- ▶ Last day to change modules

Seats in the Lab

Either a dedicated seat with a bookshelf and a cupboard, OR a “hot desk”.

Rules of Conduct

1. Be considerate to other people in the lab.
2. Keep it neat, and clean up after yourself.
3. You may eat and drink in the lab, but see Rule 2.
4. Don't ever jam the outside door (to the 5th floor Engineering lecture halls) open.
5. Coffee is R5 per cup; mark it on the sheet, and pay at the end of the month.

MENTAL HEALTH ADVICE

Although COVID-19 restrictions still apply, we strongly urge you to work in the lab whenever possible and regulations allow.

- ▶ Is a compulsory 32-credit year module
- ▶ Consists of a substantial piece of independent research or software engineering
- ▶ Typically involves writing either nontrivial or a large amount of code (or both)
- ▶ Will be supervised by one or two lecturers
- ▶ If done well, is an excellent addition to your resumé...
and helps when looking for an MSc supervisor
- ▶ Has the following deliverables at the end of the year:
 1. a project report
 2. a project demonstration
 3. a project presentation
- ▶ Ultimately, puts you in the driving seat

The module framework is available at cs.sun.ac.za/courses/cs771

PROJECT SCHEDULE & MARK WEIGHTS

DATE	TYPE	DESCRIPTION	WEIGHT
2 Mar	F	Supervisor may require a 2-page “project brief”	—
25 Mar	F	Max. 7 page draft report; see the module framework	—
20 May	S + F	DEMO 1 Max. 7 page report (as above) + demo*	10%
18–22 Jul	S	DEMO 2 Just demo; no report	20%
2 Sep	F	Draft final report	—
21 Oct	S	DEMO 3 Final report	30%
7–11 Nov [†]	S	DEMO 3 Final demo	30%
[ditto]	S	DEMO 3 Presentation	10%

F = formative (i.e., for feedback, whether written or via meeting)

S = summative (i.e., for marks)

*To be scheduled in the week of 23–27 May, but may be the next week

[†]May be moved one week earlier or later—subject to lockdown level and moderator availability

Software Engineering vs Research

Is this a well-defined project that mainly involves building a piece of software?

OR Is the goal to (in)validate some hypotheses?

OR Is it a bit of both?

Bottom Line

- ▶ There is only one prescribed report \LaTeX style, but the contents differ somewhat:

§	SOFTWARE ENGINEERING	RESEARCH
1.	Introduction	Introduction
2.	Background & requirements	Literature survey & related work
3.	Design & implementation	Hypotheses & experimental design
4.	Testing	Results & evaluation
5.	Conclusion & future work	Conclusion & future work

- ▶ **These are just broad ideas... always listen to your supervisor!**

You

- ▶ MUST schedule regular project meetings with your supervisor
- ▶ MUST use your assigned CS GitLab repositories
- ▶ MUST submit all reports for summative assessments via SunLearn
- ▶ SHOULD work on your project during recess
- ▶ **MUST work hard from Day 1, at least 10 hours per week (official number)**

Your Supervisor

- ▶ MAY have to deviate from meeting schedules because of other responsibilities
- ▶ MAY require you to follow a software engineering process (→ RW344)
- ▶ MAY require regular progress reports or a backlog
- ▶ MAY deviate from these rules, but **MUST NOT** contradict the module framework

Deliverables & Marking Scheme

Report Marks are based on the quality (not quantity) and clarity of the report, and the scope of the work.

Demo Show code running (app, notebooks, etc.); marks are based on the quality and scope of the work, and insight when answering questions.

Presentation Marks are based on the quality and clarity of the presentation.

There is a 2nd opportunity for Demo 3, and we won't hesitate to fail students on the 1st opportunity. (Then graduation in December is impossible.)

Postgraduate Studies: Your First Taste of "Real Life"

- ▶ Use to your benefit what you're good at, and where you can improve.
- ▶ Deliver an excellent product.
- ▶ Convince us (report, demo, presentation).

WHAT WE'D LIKE VS REALITY

IN OUR MINDS

You will start working from Day 1.

You will work at least 10 hours per week, every week.

You will make regular progress.

You will work regularly throughout the holidays.

You will finish with time to spare, to polish your app and/or report.

REALITY

You will wait too long to start.

You have other modules and deadlines, and the project takes a back seat.

You *will* get stuck at least once.

You go spelunking, binge on series, or disappear into gaming.

You run out of time, and try to do everything the week before a deadline.

- ▶ Schedule your work time wisely.
 - ▶ Block daily project time, just like for any other subject.
 - ▶ Remember, context switching is expensive.
 - ▶ Keep a list of project things you can do with half your neurons on vacation, like refactoring your code, or making graphics for your report.
- ▶ Schedule regular down time, daily and weekly.
 - ▶ Make time for whatever keeps you sane.
 - ▶ Try not to pull all-nighters.
 - ▶ By all means, take a couple of days totally off during recess... but if this runs beyond two weeks during the winter recess, you'll probably end up in trouble.
- ▶ Keep backups of everything. Commit your code regularly.
- ▶ Academic writing takes longer than you expect, even if you've done it before. Start early, and expect to write and rewrite.
- ▶ If there is an issue, speak up. Immediately.

Your Choice

1. Read through the projects posted.
2. Discuss the projects you find promising with their supervisors.
3. You may propose a project topic, but
 - ▶ you need to find a willing supervisor, and
 - ▶ each supervisor can only supervise one student-proposed project.
4. By **Friday, 11 February**, rank your 7 top choices via the link given on Slide 4.
 - ▶ You **MUST NOT** choose more than 3 projects from the same supervisor.
 - ▶ You **MUST** choose from at least 4 different supervisors.

Allocation

- ▶ We strive to make a fair assignment, but we cannot guarantee you will get any of your top three choices.
- ▶ The project allocations will be published on **Friday, 18 February**.

- ▶ You **MUST** offer 96 credits = 6 CS Honours modules.
- ▶ You **MAY** offer at most 2 modules from related departments.
 - ▶ Math and Applied Math are typical choices and work well, depending on your academic strengths and history.
 - ▶ But be careful: Such modules have their own prerequisites and different timetables, and some modules may be worth fewer credits.
- ▶ We *strongly urge* you to **balance your choice to 48 credits** (= 3 CS Honours modules) **per semester**; if you do overload, frontload the first semester.

To Do

- ▶ Make your choice by **10 February** (necessary for registration).
- ▶ If you are uncertain, attend the first week's contact sessions for those modules in which you are interested.
- ▶ Any changes to your initial choice **MUST** be finalised by **25 February**.

FIRST SEMESTER

64947-712	Advanced Algorithms	Willem Bester
64955-713	Advanced Automata [Theoretical Computer Science]	Lynette van Zijl
64963-714	Concurrent Programming RW314	Cornelia Inggs
64971-716	Computing & Society [Advanced Topics I]	Bill Tucker
14195-742	Machine Learning A RW315	Steve Kroon
65021-745	Compilers [Software Construction]	Bernd Fischer
14232-791	Artificial Intelligence	Andries Engelbrecht
64572-793	Digital Image Processing APPLIED MATH	Milton Maritz

SECOND SEMESTER

63452-711	Automata Theory & Applications RW345	Brink van der Merwe & Lynette van Zijl
11788-741	Machine Learning	Andries Engelbrecht
65013-744	Software Testing & Analysis [Concurrent Programming II]	Cornelia Inggs, Willem Visser & Bernd Fischer
65048-745	Principles of Data Science [Advanced Topics II]	Marcel Dunaiski
14066-791	Space Science Algorithms	Trienko Grobler
62847-792	Computer Vision APPLIED MATH	Willie Brink
13944-795	Functional Programming	Brink van der Merwe
14065-796	Vulnerability Discovery & Exploitation [Software Verification & Analysis] TBC APRIL	Fabian Yamaguchi
