Research School of Computer Science

Student Handbook

Research School of Computer Science
College of Engineering and Computer Science
The Australian National University
CANBERRA ACT 0200

Phone (02) 6125 4043
Fax (02) 6125 0010

Preface

This handbook describes the administrative and practical arrangements that students need to be aware of in order to study effectively in the Research School of Computer Science, ANU College of Engineering and Computer Science. Our School is the ANU hub for education in the professional, technical and scientific aspects of computing. In particular, our School has educational strengths in fundamental computer science, computer systems, software engineering, artificial intelligence and human-centred computing. The School offers many courses in several degree programs across these areas.

A world-class University such as the ANU is a research-intensive institution. The strong research activities in the College make it possible for students to participate in a research-led education, and you can expect to be taught by staff of high research achievement during your degree program.

Things change fast in computing, and your program structure and its component courses will change during your studies to reflect leading-edge changes in professional practice and in knowledge and invention in the discipline. This means that you need to check for changes in the courses we offer each year as you progress through your program.

This handbook is the work of many people in the Research School of Computer Science. We welcome feedback on what we have left out and what can be improved.

Professor Alistair Rendell
Director
Research School of Computer Science

Contents

Chapter 1: Introduction

General aims

The teaching programs in the Research School of Computer Science (RSCS) address a professional education in software engineering and information technology as well as a scientific education in computer science.

The Research School aims to produce graduates with a four-year, professional education in Software Engineering (the Bachelor of Software Engineering), a four-year technical and professional education in Computer Science (the Bachelor or Advanced Computing), or a three-year technical and professional education in Information Technology (the Bachelor of Information Technology). We support the study of information systems in the Bachelor of Commerce and of computer science in the Bachelor of Science, and we also provide computing courses for students from all areas of the university. We offer an elite four-year program in computing: the Bachelor of Advanced Computing (Research & Development). Graduates of our various Honours programs, including the BIT (Hons), have been accepted for postgraduate studies at leading computer science departments around the world.

We also offer a two-year Master of Computing program that provides students with exposure to best practice in computing with cutting edge courses in areas of relevance to computing professionals

Our teaching programs are held together by the core discipline of computing. In the following section, we shall briefly describe this discipline.

The nature of the computing discipline

Computing is a new discipline in academic terms. The oldest computing Schools are about 50 years old, while the great majority of Schools have been established in the last thirty years. At the ANU, for example, the old Department of Computer Science was formally established in 1976. In 1998 ARG (Automated Reasoning Project) merged with Computer Science Laboratory (CSL). The new group became the computer science arm of the Research School of Information Science and Engineering (RSISE). In 2009 the Computer Science Laboratory and Department of Computer Science merged to create the School of Computer Science and was renamed as the Research School of Computer Science in 2011.

Awareness has steadily emerged that the computing discipline really is different to the older academic disciplines, and that some (but not all) students find it hard.

This was considered in an influential report (see the report “Computing as a discipline” of the Association of Computing Machinery (ACM) Task Force on the Core of Computer Science, Communications of the ACM, 32/1, Jan 1989, p. 9; available on the web through the ACM Portal) which identified the following paradigms and forcefully argued that all three are of equal importance to the computing discipline.

Theory

This has its roots in mathematics. It involves the derivation of properties of certain defined objects of study, using the methods of mathematics and logic.

Abstraction

This has its roots in engineering, the natural sciences, and in the experimental scientific method. It involves the identification of an abstraction or model. Predictions of this model are made and tested by experiment.

Design

This has its roots in engineering. It involves the design and construction of a desired system and its subsequent testing.

To quote from the ACM report:

Computing sits at the crossroads among the central processes of applied mathematics, science, and engineering. The three processes are of equal--and fundamental--importance in the discipline, which is a unique blend of interaction among theory, abstraction and design. The binding forces are a common interest in experimentation and design as information transformers, a common interest in computational support of the stages of those processes, and a common interest in efficiency. (p. 11)

Put another way, the mastery of the computing discipline involves working with a wider range of paradigms or ways of thinking than with any other academic discipline.

The broadening scope of computing and the enormous and rapid changes it is bringing to the world mean that the courses that are the "right" ingredients of a university degree program in computing are always changing. The courses that you will see on offer when you start a three, four, or five-year course will change maybe two or three times to keep the program up to date.

Is computing hard? Is studying computing hard?

It seems clear that some people find it hard. The following observations might be relevant as to why this is so.

It is an empirical fact that some computing professionals are ten times more productive than the average. Such an extreme range of productivity is said to be very unusual in other professions and to be unique to computing. No adequate explanation has emerged, but it is thought to be related to the multi-paradigm nature of computing just indicated.

Similarly it is an empirical fact that some students can complete a programming assignment much faster than other students who are also very competent. This means that where some students might take ten hours to complete an assignment, other students might take fifty hours to complete it to the same level of performance. The problem is not peculiar to computing at the ANU, but applies to computing departments across Australia and the world.

If this situation is allowed to remain, then computing studies can cause a workload blow out. In relative terms, computing courses will take far more time than other courses, and there is not enough time in the study week to properly address the other courses.

Rather than being too hard, then, perhaps the question is: "does the study of computing take too much time?" As teaching academics we engage in more debate (and are subject to more criticism) over this question than any other.
It is clear that

- We, the academics, must adopt strategies to contain workload blow out, and that
- You, the students, must adopt strategies to contain workload blow out.

Such strategies are discussed in several places later in this handbook, and are a focus of the interaction between academic staff and students.

Chapter 2: School Services

Introduction

Students need to understand what the School is responsible for and what it is not responsible for. We are happy to accept responsibility for the former, but not for the latter.

In general terms, the School teaches a range of computing courses at first-year, second year, third year, fourth year and Masters Level. The students who take these courses are enrolled in a wide range of programs, and count the courses towards their programs. The School is responsible for these courses, for their content, and for reporting the results.

The School is *not* responsible for the programs that the courses are counted towards. Rather, a program is administered by the relevant area of the ANU. For example, if you are enrolled in the BSc degree program then the question of whether a particular computer science course can be counted towards your program must be addressed to the College of Science Office. BAC, BIT and BSEng programs are handled by the College of Engineering and Computer Science; BComm by the College of Business and Economics.

The School is *not* responsible for admission to the ANU or for admission to any particular program. Rather, we see students after they have been admitted to the university and have been enrolled in a particular program.

Course level services

In each course the course convenor should make a clear statement about the services that the course provides. Generally the School goes to some considerable trouble to set up small-group sessions. These are either tutorials; where a tutor works with a group of perhaps 20 students, or programming laboratory classes, where a demonstrator works with a group of perhaps 20 students. These classes are your usual opportunity to ask questions. In addition, a course might have an electronic question-and-answer service, based on an electronic bulletin board or e-mail (although this is up to the lecturer).

At times, of course, it is appropriate to see the course convenor. Typically the convenor will report 'office hours', during which he or she is available without appointment, or will provide a mechanism for making an appointment.

School-level services

There are issues, concerns and grievances that you will want to discuss with a School representative rather than your course convenor. At times you may want to discuss an issue with the Director of the School. Appointments should be made through the School Office. The office is open Monday to Friday from 9am to 5pm, apart from breaks. The phone number is 02 6125 4043 (54043 internally), fax number is 6125 0010 and you can e-mail to admins.cs.cecs@anu.edu.au

College-level services

The Associate Dean (Education), of the College of Engineering and Computer Science (Assoc. Prof. Paul Compston) is a significant position, in that he is the *prescribed authority* for all matters concerned with your enrolment and progress towards the degree.

Within the School the Associate Director (Education) Dr Ramesh Sankaranarayana has delegated authority and provides overall coordination for coursework.

Associate Director Education Coursework, Semesters 1 and 2: Dr Ramesh Sankaranarayana
ramesh@cs.anu.edu.au

The Associate Director asks that students consult one of the program convenors on all matters of program enrolment and progress.

Students can contact the Student Administrator requesting contact details of the relevant Program Convenor or send an email to the address(es) below:

- Convenor for Bachelor of Information Technology: Professor Tom Gedeon tom@cs.anu.edu.au
- Convenor for Bachelor of Software Engineering: Dr Shayne Flint shayne.flint@anu.edu.au
- Convenor for Bachelor of Advanced Computing: A/Professor Peter Strazdins peter.Strazdins@cs.anu.edu.au
- Convenor for Bachelor of Advanced Computing (R&D): A/Professor Peter Strazdins peter.Strazdins@cs.anu.edu.au
- Single-semester Projects Co-ordinator: A/Professor Weifa Liang weifa.liang@anu.edu.au
- Convenor for Honours in I.T.: Professor John Slaney john.slaney@anu.edu.au
- Convenor for Coursework Masters: Lynette Johns-Boast lynette.johns-boast@anu.edu.au

Student Administration

For all administrative procedures relating to students of the College of Engineering and Computer Science and your enrolment at the University such as:

Academic Progress
Advanced Standing
Appeals
Graduation
Program advice
Program leave
Program transfers
Scholarships
Welfare

See the College Student Advisor
Bindi Mamouney
Monday – Friday 8.30am –4pm
Level 2, CSIT Bldng 108, room N202
T: 6125 8870
E: student.services@cecs.anu.edu.au

Is there democracy?

The short answer is “yes”.

All students have representation on the Student Representative Committee (SRC) and representation on the CECS Education Committee.

The current membership of the SRC Committee (http://cs.anu.edu.au/study/currentstudents/src) is:

- 2x 1st year representatives
- 2 x 2nd year representatives
- 2 x 3rd year representatives
- x 4th year representatives
- 2 x Masters representatives
- 2 x HDR representatives
- Computer Science - Director of School or Delegate
- Computer Science – School Manager
- Computer Science - Student Administrator

Nominations for membership of the SRC will be called for in the first few weeks of Semester 1 with an election where necessary.

The SRC meets two times per year as the committee only, and two times per year as part of the School Staff meeting. This committee provides an opportunity for discussion on any and all student matters.

The current membership of the CECS Education Committee is:

- Associate Dean Education
- Associate Director Education (both Schools)
- Associate Dean (HDR)
- Education Dean
- Dean, CBE
- Director, CPMS
- Directors (both Schools)
- Manager, Student Services
- Educational Developer
- ANUSA Student Representative

CECS Education Committee usually meets regularly throughout the year. This Committee provides advice to the Executive Committee on strategic planning for improvements in educational excellence on a range of matters.

Chapter 3: CSIT Building

General information

In early 1995 the Research School of Computer Science moved into the north wing of the Computer Science and Information Technology (CSIT) building #108 (map reference GH54).

In approximate terms the use of the CSIT building is

Level 1 Student facilities, including programming laboratories, tutorial rooms, and Masters lab.

Level 2 Staff and research facilities, academic and professional staff offices, machine room, project laboratories, and postgraduate coursework laboratories.

Level 3 Staff and administrative facilities including School Reception, academic and professional staff offices, and graduate student corrals.

The normal building opening hours are 9.00 a.m. to 5.00 p.m. Outside these hours students will have certain access to the facilities on Level 1 through the Student Entrances, controlled by the student card. Students have no access to the rest of the building. It is expected that 24-hour access will be provided to a subset of the programming laboratories.

A floor plan of level 1 is provided over the page. The telephones in labs are only used for internal calls or for external calls to free numbers.

If access does not work:

1. Check with Security (x52249)
2. email admins.cs.cecs@anu.edu.au explaining the card is not working and access is required, including your name and university I.D.

Building regulations

You should understand that the Australian taxpayer in general and the ANU in particular have gone to a lot of expense to provide teaching facilities in the CSIT building. This is done on an implied contract that there is a common sense and sensible use of the building by academics and students.

The pragmatics of the situation, however, is that we cannot rely on an unstated contract. At times explicit regulations might be posted, such as:

- **No smoking** is permitted in the building; and
- **No eating** or **drinking** is permitted in laboratories or tutorial rooms.

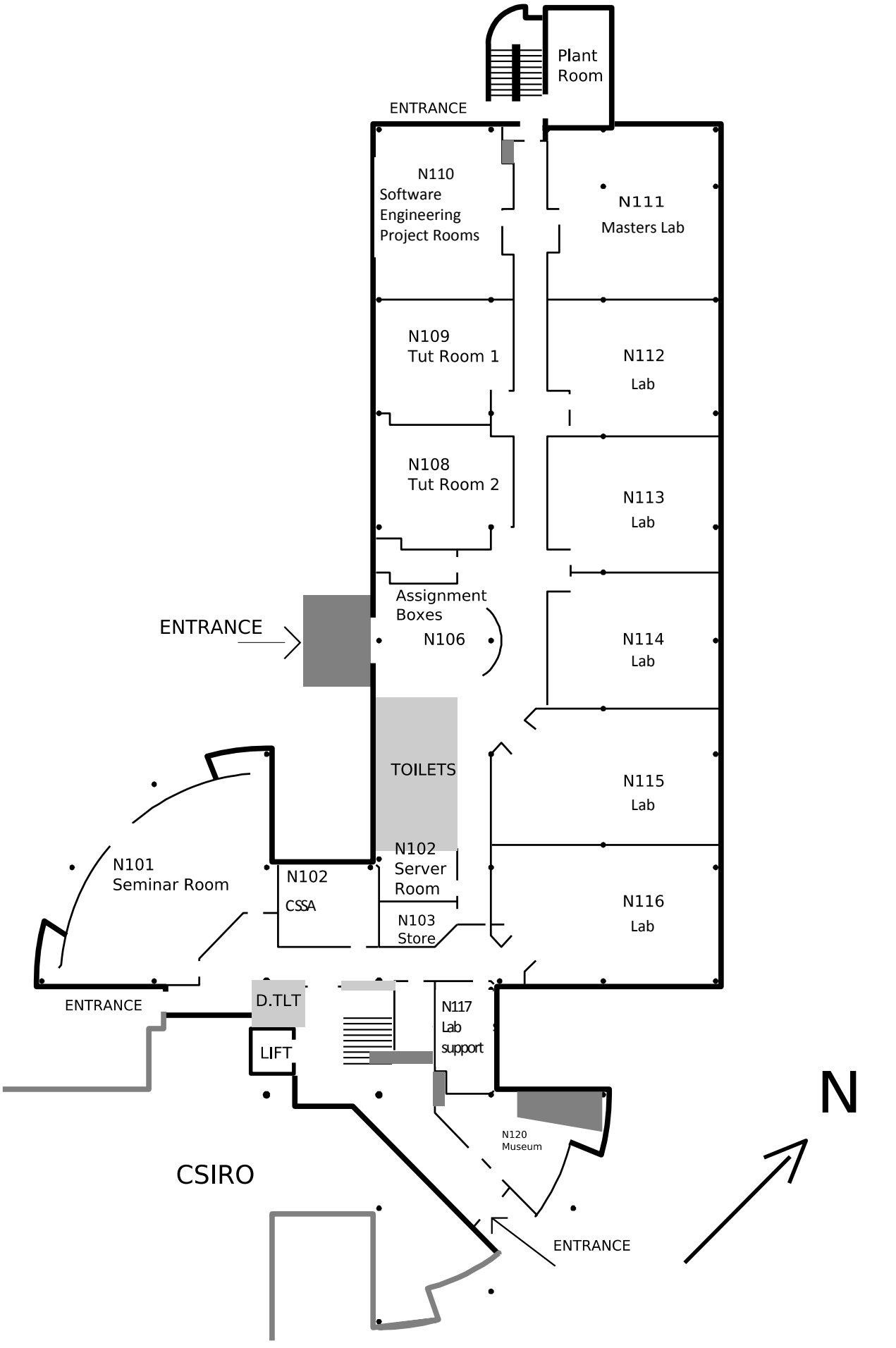
At other times, the regulations may be implicit, such as:

- Equipment must only be used for its intended purpose;
- Equipment must not be moved from its assigned location;
- Furniture must only be used for its intended purpose; and furniture must not be moved from its assigned location (in particular, gas lift chairs must not be taken out of computer labs).

Building regulations, both explicit and implicit, must be considered to be in force. If situations arise where building usage is unacceptable, for instance, where

- equipment and furniture is broken, or
- cleaners must cope with areas that have been trashed,

Where the perpetrators are unknown, then the general amenity of the building cannot be guaranteed. The School's budget for replacement furniture is not bottomless and the cleaners' patience for cleaning up mess is not endless. In these cases, then, the behaviour of a few might cause the inconvenience of the many.



The *A*

CS&IT Building: ANU Wing, Level 1

Chapter 4: Teaching Programme

As mentioned in chapter 1, the fast rate of change in the computing discipline means that our degree programs also change frequently. Degree programs also allow for various courses to be taken from other Schools.

Courses

Courses taught by the Research School of Computer Science in 2014 are the following. See the programsandcourses (http://programsandcourses.anu.edu.au/) pages for more details. In rare circumstances courses will be unable to be run due to the lack of resources. In these cases students will receive notification well in advance.

The School teaches the following first-year courses:

COMP1100	Introduction to Programming and Algorithms
COMP1110	Introduction to Software Systems
COMP1130	Introduction to Advanced Computing I
COMP1140	Introduction to Advanced Computing II
COMP1510	Foundations of Software Engineering
COMP1710	Web Development and Design
COMP1720	Art Interaction & New Media
COMP1730	Programming for Scientists

The School teaches these second-year courses: (some of these may be taken by students in their first year)

COMP2100/2500	Software Construction /for Software Engineers
COMP2130	Software Analysis & Design
ENGN2219	Computing for Engineering Simulation
COMP2300	Introduction to Computer Systems
COMP2310	Concurrent and Distributed Systems
COMP2400	Relational Databases
COMP2410	Networked Information Systems
COMP2550	Advanced Computing R&D Methods
COMP2560	Studies in Advanced Computing R&D
COMP2600	Formal Methods in Software Engineering
COMP2610	Information Theory
COMP2620	Logic

The School teaches these third-year courses:

COMP3006	Computer Science Research Project
COMP3100/3500	Software Engineering Projects (annual)
COMP3120	Managing Software Development
COMP3130 [odd years]	Computer Science Research Project
COMP3300 [even years]	Operating Systems Implementation
COMP3310 [even years]	Computer Networks
COMP3320 [even years]	High Performance Scientific Computation
COMP3420	Advanced Databases and Data Mining
COMP3530	Systems Engineering for Software Engineers
COMP3550	Advanced Computing R&D Project

COMP3600	Algorithms
COMP3610 [odd years]	Programming Languages
COMP3620	Artificial Intelligence
COMP3630 [even years]	Theory of Computation
COMP3700/3710	Topics in Software Engineering/Computer Science
COMP3740	Project Work in Computing
COMP3820	Software Engineering Internship
COMP3900	Human Computer Interface Design and Evaluation

It teaches these fourth-year courses:

COMP4005 P/F	Computer Science IV Honours
INFT4005 P/F	Information Technology IV Honours
COMP4130	Managing Software Quality and Process
COMP4300 [odd years]	Parallel Systems
COMP4330 [odd years]	Real-Time & Embedded Systems
COMP4500	Software Engineering Practice
COMP4540	Software Engineering Research Project
COMP4550	Advanced Computing Research Project
COMP4560	Advanced Computing Project
COMP4600	Advanced Algorithms
COMP4630 [odd years]	Overview of Logic & Computation
COMP4650	Document Analysis
COMP4660 [even years]	Bio-Inspired Computing
COMP4670	Introduction to Statistical Machine Learning
COMP4680 [even years]	Advanced Topics in Statistical Machine Learning
COMP4710	Topics in Software Engineering
COMP4800	Industrial Experience

Graduate courses are:

COMP6240	Relational Databases
COMP6260	Formal Methods in Software Engineering
COMP6262	Logic
COMP6300	Introduction to Computer Systems
COMP6310	Concurrent and Distributed Systems
COMP6311	Software Analysis and Design
COMP6320	Artificial Intelligence
COMP6330 [even years]	Operating Systems Implementation
COMP6331 [even years]	Computer Networks
COMP6340	Networked Information Systems
COMP6353	Systems Engineering for Software Engineers
COMP6361 [odd years]	Programming Languages
COMP6363	Theory of Computation
COMP6390 [even years]	HCI and Usability Engineering
COMP6442	Software Construction
COMP6463 [odd years]	Overview of Logic & Computation
COMP6464 [even years]	High Performance Scientific Computation
COMP6466	Algorithms
COMP6467 [odd years]	Introduction to Statistical Machine Learning
COMP6470	Special Topics in Computing
COMP6490	Document Analysis
COMP6700	Introductory Programming
COMP6710	Introduction to Software Systems

COMP6719	Computing for Engineering Simulation
COMP6720	Art and Interaction in New Media
COMP6730	Programming for Scientists
COMP7310 [Online]	ICT Sustainability
COMP8100 [odd years]	Requirements Elicitation and Analysis
COMP8110 [even years]	Managing Software Project in a System Context
COMP8173 [odd years]	Software Engineering Processes
COMP8180 [even years]	Systems and Software Safety
COMP8190	Model-Driven Software Development
COMP8400	Algorithms and Techniques for Data Mining
COMP8420 [even years]	Bio-inspired Computing
COMP8440	Free & Open Source Software Development
COMP8460	Advanced Algorithms
COMP8600	Introduction to Statistical Machine Learning
COMP8650	Advanced Topics in Statistical Machine Learning
COMP8740	Artificial Intelligence Project
COMP8750	Computer System Project
COMP8780	Information and Human Centred Computing Project
COMP8790	Software Engineering Project
COMP8800	Computing Research Project

Student & Course Websites

The main website for current students is http://programsandcourses.anu.edu.au/. The site contains up to date resources on things you need to know about your degree program, courses and what is available in the current year.

Traditionally, each of our undergraduate/masters courses has a home webpage on the School's student directory, e.g. at http://cs.anu.edu.au/student/comp1100 for COMP1100. The hub page for all course web pages is http://cs.anu.edu.au/courses. Lecturers use this course website to post a detailed course description, and it may also be used for all of the notes, assignments, and bulletins.

In recent years the University's Wattle system has been increasingly used for hosting our (and other courses) on campus. You may find that, as the semester progresses your attention will be increasingly drawn to this portal for your computing courses.

There is also a student portal on Wattle which includes an announcement forum.

Students are expected to check their course websites regularly. Many courses also have an interactive forum for announcements and discussion, either through Wattle https://wattle.anu.edu.au/ or via the StReAMS interface https://cs.anu.edu.au/streams/.

Chapter 5: Enrolment and Registration

5.1 Introduction

In the Research School of Computer Science we use the term *enrolment* to refer to the process relating to choice of programs and courses, and *registration* to refer to the process relating to choice of tutorials and laboratory groups. Students must do both parts of the process for enrolled courses.

It is *your* responsibility to ensure that you are formally enrolled in the courses you are intending to study during the academic year. It is also your responsibility to enrol in courses which are consistent to your degree's rules. You enrol in courses on-line through ISIS (Interactive Student Information System). You do *not* enrol with the Research School of Computer Science. See http://students.anu.edu.au/manage/enrol/standard.php for more information on enrolment.

The only time you need to consult the School for course enrolment is when you do not have the prerequisites for enrolment in a particular course. Under some circumstances the relevant Associate Director, i.e. CS or Eng, may give permission to enrol without the prerequisites, and the College Student Office will then complete the enrolment. If you wish to apply to enrol without a prerequisite you will need to discuss the situation with the relevant course convenor and obtain their written approval (an email is sufficient) then complete an Enrolment Variation Form.

Most of you will have enrolled in computing courses (see Chapter 4) during enrolment week. You have up until the second Friday in the semester to change your enrolment. The Research School of Computer Science has no quotas in these courses. You do *not* need permission from us to enrol in first-year courses.

In the meantime it is *recommended that you follow the guidelines* outlined below, at the start of the academic year. The University has enrolment procedures that must be met in order to meet government legislation in the form of census dates. To avoid financial and academic penalty it is recommended that you become familiar with important dates in the academic calendar.

5.2 Registration with the School

The most important information is your course enrolments for the coming semester. The School gets this information automatically from the University, BUT: *no matter how long ago you paid your fees or enrolled in subjects through ISIS, the School only gets to see this information from Student Administration after you collect your student card!* So, the best way to speed things up is to collect your new student card, or have your existing student card re-validated, as soon as possible. On the morning of the next business day after collecting your card, we should have your RSCS course enrolment data. See http://students.anu.edu.au/manage/student-cards.php for more information on student cards.

Once we get your RSCS course enrolment data, and you login to StReaMs a computer account will be created (or reactivated, for later year students) for you on the RSCS student systems. To login to the account, you will use your University ID as the username, and your current campus-wide PAC (Personal Access Code) as your password. You can change your PAC using the ISIS system or by visiting a Student Consultant at the InfoPlace in the Chifley Library.

5.3 Class Timetables

The University timetable system schedules the lectures for nearly all courses. You should ensure that you are able to attend all of the lectures in your courses (unless they are marked as "repeat" lectures).

The timetable system also schedules tutorials and laboratory classes. These are small group classes and you have some choice of which class you attend. In the Research School of Computer Science we provide a powerful automatic booking system called StReaMS that you must use to register your choice.

During Orientation Week students should register for tutorials and/or laboratories for all their computing courses. Registration will generally be available until the end of week 1 (first teaching week of semester), but classes fill up on a first-come first-served basis, so early registration is strongly advised.

Registration for courses is made using a computer system called StReaMS ('Student Registration and Marks System') to record your choices. To log in, connect through the web at http://cs.anu.edu.au/streams, and use your University id and PAC code to log in. The computer laboratories on Level 1 (ground floor) of the CSIT building will be available during Orientation Week for StReaMS registration – use the account `guest' (with the same password) if you have not logged into StReaMS before.

In selected laboratories, there are notices explaining how to register. There are also notices giving the tutorial and laboratory times for each course. To save time, think about which classes you prefer *before starting the registration process*.

When you use the registration system, make sure you have your student number, PAC (personal access code), and timetable with you.

5.4 What else do you have to do?

The standard University arrangement is that a wide range of teaching material is made available as 'handouts', either as a collection at the start of the semester or as needed through the course. Many handouts are available only from the course websites. Large documents, such as lecture notes or computer manuals, are not included in this arrangement but need to be purchased by you.

Most courses in Computer Science provide all the extra material online through the course website, at no charge.

5.5 Lecture timetable and locations

You should consult the *University timetable* for lecture times. See the university timetable web page http://timetable.anu.edu.au. If you have a clash of class times, tell the course convenor as soon as possible (email or in person).

Chapter 6: Assessment in Computer Science Courses

Assessment is the focus of a great deal of effort and consumes a great deal of nervous energy. This section attempts to place assessment in perspective, and to bring together some of the issues that are relevant to the assessment process.

6.1 A global look at assessment

The School describes the objectives of its courses in terms of the skills that a student will gain from the course. A typical statement is:

At the completion of this course the student will be able to;

- *Understand the theory and practice of several methods of expressing and managing concurrency,*
- *Read and write programs in the C language in the Unix programming environment,*
- *Read and write programs in a concurrent programming environment.*

When the Chair of Examiners awards a student a pass grade in a course he or she is attesting to the university and to future employers that the student does in fact have those skills. This judgement is based on assessment.

The assessment categories, and some of their more obvious characteristics, include the following.

Formal examinations

The end-of-course examinations fall into this category. They are run by the Examinations Section **not** by the School. They are carried out under strict invigilation and attendance is carefully recorded. For most courses they are two or three hours in length. They are one of the fairest ways of evaluating a student's competence in a course. You are required to be available during any examination periods as listed in the ANU Principal Dates. http://about.anu.edu.au/principal-dates

Class examinations

Mid-semester examinations fall into this category. The academic staff responsible for the courses conducts these exams. Typically they are conducted in a timetabled lecture session, and are one hour in length.

Assignments

Assignments involve a task specified by the lecturer, with a submission by the student in a nominated format. They may be computer programming, descriptive writing, or mathematical analysis.

Class examinations and assignments are progressive or ‘continuous’ assessment, in that they are completed part way through the course. The most important way for students to get feedback on their learning is to look at the marked assignment or class exam paper, and lecturers and tutors will often put a lot of effort into marking papers with comments as well as numerical marks. The final examination at the conclusion of the course determines whether a student has achieved an acceptable competence in the subject matter, and these exams provide detailed feedback.

In each course the course convenor is responsible for determining the system of assessment. Convenors must explain and discuss the assessment system with students, but the convenor decides

what it will be. The assessment scheme for the course will be published in the first week of the semester and discussed in classes in week 2. Any changes after week 3 will be rare. Note that the mapping of numerical scores to grades in the initial assessment specification is approximate. The final marks may be scaled following the end of semester Examiners meeting.

The overall result for a course is subject to Examinations *(The Faculties) Rules*, which can be found on the ANU website. A quick reference version is at: http://www.anu.edu.au/sas/examinations/index.php
You should read these rules at least once during your university career.

The College of Engineering and Computer Science has a formal appeal mechanism which applies to the case where a student wishes to dispute the mark awarded for a course or the mark awarded for any component of the assessment for a course. This is described in Appendix A of this Handbook.

6.1.1 Formal examinations

Formal examinations essentially have a certification role. They are intended to attest that the student has reached an acceptable standard in the material of the course. They will typically consist of questions across the full syllabus of the course. They are stressful occasions but are considered by many to be the fairest method of final assessment.

There is, under normal circumstances, *no feedback* to the student about the final examination apart from the mark. You may seek to discuss your paper with the convenor, if you think that such a discussion will be fruitful, but you are not permitted to contact the convenor until after you are officially notified of the result for the course. You are allowed to read through your marked exam paper later by making an appointment.

It is your responsibility to ensure your availability in the Canberra region to sit any kind of Formal Examination in any of the courses that you enrol in. This includes mid-semester examinations (if applicable to the course; usually held in week 7 or week 8). It also includes Special or Supplementary Examinations; which you should expect to be held in the first week of the following semester. For this reason, you should not make any (unalterable) travel arrangements to be absent from Canberra during these times.

Students should note the School's policy on Special Consideration and Special Examination, and procedures for their application. In particular, if you feel unwell on the day of an examination, you should apply for Special Consideration unless the symptoms are severe and can be verified objectively, as a Special Examination will only be granted under extraordinary circumstances. Please see the Current Students webpage for details on the policy and procedure.

6.1.2 Class examinations

An essential characteristic of a class examination (commonly called a Quiz) is that it provides feedback to the student. A class examination is intended to give you an idea of what you do know, and to give you an idea of what you are expected to know but don't know. Typically you will get your examination paper returned to you with marks and perhaps annotated as to where you were wrong.

6.1.3 Assignments

An assignment involves the application and the extension of the material and the concepts that have been presented in lectures and textbooks. This application should cause you to think through the issues involved and come to a better understanding than you would otherwise. Assignments,

particularly programming assignments, have an operational and a practical aspect to them. The engineering aspects of computing are about actually *constructing* complicated beasts called program systems. This construction cannot be learned from a book but must be learned through laboratory experiment and experience.

The details of what is required for an assignment and the format of its submission will vary between courses. Some may require electronic submissions; some may only take written or printed submissions. You should carefully note the format required before you start work. Many assignments will require a report, not merely a program. Some guidelines for writing reports are found at this website https://academicskills.anu.edu.au/taxonomy/term/130

Your assignment submission is normally marked by your tutor. A key issue about assignments is the feedback that the tutor can give you about the quality of the understanding and constructive skills that are represented in the assignment submission. You will get your submission back, marked and annotated. A discussion of the assignment issues may be used as the focus of a subsequent tutorial session. The feedback is completed when you apply the ideas discussed in your next assignment and demonstrate them in an examination.

We undertake to:

Return your submission in a timely fashion. Our objective is to return the material generally within three weeks of the submission deadline;

- Provide appropriate feedback. This may take the form of annotations to your submission, the provision of a ‘standard solution' to the problem, a discussion in a tutorial session, or a combination of these;

- Carefully monitor the workload of the assignment components. Each assignment specification will involve an estimate of the time that we think the assignment will take and you are asked to state with your submission how long it actually took. This will help us with our estimates and with possible revision of the courses assessment scheme (for example, if an assignment turned out to be much harder than expected then it might be given a greater weight).

Generally there will not be any large assignments due after week 12 in any semester.

6.2 Student workload

We expect an “average” student to spend about 130-150 hours working on a 6 unit course. This includes time spent in lectures and supervised laboratories, independent study and assignments. But it does *not* include the time you spend studying for your final examination and the exam itself. Here is one possible breakdown of this workload:

3 lectures/week over 12 weeks plus 1 hour of study for each hour of lecture makes 72 hours
6 labs at 2 hours with 2 hours work outside of labs makes 24 hours
final week review lectures (plus one hour study) makes 4 hours
two assignments at 30-50 hours (at 1 to 1.5 hours per percentage mark)

Of course, this breakdown will vary between courses and between students. In particular, students who are new to programming may need to spend quite some time in computer labs in order to “get the knack”. As we observed in the first chapter, some people are always slower than others to complete programming assignments.

We urge all of you to study continuously during the semester rather than waiting for the assignment and examination rush. Review your lectures each day and ask yourself if you really understand them. Take the computer laboratory sessions seriously and construct your own little lab exercises to experiment with the lecture and laboratory material. Look at sources outside of the prescribed textbook. Study the provided solutions to lab and assignment exercises and understand how and why your solutions might be improved!

We label each assignment with an estimate of the time it will take to complete. In view of the nature of the computing discipline discussed earlier (in Section 1.3), we might expect two-thirds of the class to complete the assignment in this time.

The number of seats in computer laboratories is carefully estimated ahead of each academic year. This number must necessarily include a reasonable averaging effect, and it cannot accommodate very high, peak loads. If two weeks are allocated to a programming assignment the wise student will work steadily and progressively through the two-week period. It could be quite foolish to attempt to complete the assignment on the night before the deadline, because there may be no available seats and the printers, file servers and networks might be overloaded! Also, it might take longer than you think. So, start working early and budget your time wisely.

6.2.1 Strategies for completing assignments on time

Here are two distinct ways of attacking a programming assignment.

1. By working on it until the program "works", no matter how long that takes. As indicated above, this could sometimes take a very long time indeed.

2. By spending a fixed amount of time on it, and arranging the submission to show that you have completed the essential structure of the program, and that you can clearly describe the nature of the completed program.

At times, you should consider using strategy (2). While you might not get "full" marks you may well get acceptable marks for a reasonable expenditure of effort.

Strategy (2) is not as strange as it might first seem. Perhaps you have always followed strategy (1) through your secondary studies. Part way through your university studies you might realise that you do *not* have time to complete every task in an absolute sense. Everything is done to a time budget, and every task is a compromise between what it would be nice to do and what there is time to do.

A third strategy for completing an assignment on time is to copy it, or part of it, from another student. This is a bad strategy. It will amount to serious misconduct as will be described below (6.4).

6.3 Illness and extensions

The great majority of students submit all assignments by the nominated deadlines and attend all examinations. The ANU system will accommodate late assignments and missed examinations in some *very exceptional circumstances*.

6.3.1 Provisions for illness and other exceptional circumstances

There is a whole section in *Examinations* (http://students.anu.edu.au/aep/exams/) (http://students.anu.edu.au/aep/assessment/) and *Assessments* that relate to students whose studies are affected by illness. *You should read these rules at least once in your academic career*

If you believe your academic performance in a course, during an examination or during the session/semester, has been adversely affected by illness, misadventure or other circumstances which are beyond your control, you may apply for special consideration. Requests should be made on the appropriate form available by downloading from: http://students.anu.edu.au/aep/assessment/consideration.php or obtained from the School Reception office. The completed form should be lodged with the School responsible for the course(s) for which you are seeking special consideration as soon as possible.

Supporting documentation from any relevant independent person (or authority) including health professionals and/or ANU Disability Advisers **must** accompany the completed form. The health professional must also clearly define the nature of the illness and fill in all areas, date and stamp the form before submitting to the office.

This form is also used to apply for extensions on an assignment(s). It is always better to do this before the assignment deadline if possible.

If you have a **chronic condition** (that is also diagnosable) that is likely to affect your ability to meet assessment requirements throughout the semester, and/or is reasonably likely to affect you on any given examination day, you should register your condition with the Disability Services Centre (http://disability.anu.edu.au/). This will give any subsequent request for deadline extension and/or Special Examination more credibility, plus you will only need to cite any supporting evidence with them, rather than to each lecturer of all courses affected. It might also enable you to obtain Special Examination Arrangements (e.g. extra time, quieter room).

If you experience a **traumatic personal event** during the semester (which is likely to affect your assessment), you should register your situation with the University Counselling Centre (http://counselling.anu.edu.au/). Not only can their professional help be of great assistance in such distressing circumstances, but, as for the above situation, evidence of the event need only be cited at that one point and it can expedite any resulting requests for changed assessment conditions.

6.3.2 Assignment deadlines

Assignment deadlines will be extended only in exceptional circumstances. In general, these must be both unforeseeable and beyond your control. One of these is illness confining you to bed, supported by a medical certificate, as noted above. Other reasons such as serious family matters, unexpected requirements to travel interstate, will be considered.

You should use the same form ‘Application for Special Consideration’. Send an email or phone the lecturer as soon as you can even if you cannot use the form.

The School has no fixed policy on extensions to assignment deadlines. Rather, the course convenor will provide a statement outlining how such requests are processed. Equally, the School has no fixed policy on penalties for late submissions. The course convenor for a course will provide a statement of the policy that applies to that course.

6.4 Misconduct in examinations

The University takes any form of misconduct very seriously. The University rules include *The Discipline Rules. You should also read these rules at least once in your academic career*.

http://www.anu.edu.au/cabs/rules/ExamsRules.pdf

The basic rule is that “misconduct” may cause suspension from the University, cancellation of a grade for a course or cancellation of a degree.

One form is “misconduct in relation to an examination.”

Here ‘examination’ means any component of assessment (examination or assignment). Further, ‘misconduct’ is defined to include:

cheating;
plagiarism, including copying; whether with or without the knowledge or consent of that other person;
except with the approval of the prescribed authority, submitting for an examination any work previously submitted for examination;
failing to comply with the University's instructions to examination candidates at, or in relation to, an examination; and
acting, or assisting another person to act, dishonestly in or in connection with an examination.

The Rule also includes a very detailed statement of the procedure that is to be followed in a case of misconduct, including appeal mechanisms.

Misconduct in examination is a serious offence indeed, and one that is taken very seriously by the School. The penalties that can be imposed under these rules are correspondingly serious. A typical penalty is that enrolment in the course is cancelled.

6.5 Collaboration versus misconduct in assignments

The computing discipline has a very strong experimental component, and this aspect of the discipline is addressed through constructive assignments, where you actually learn experimental and constructive skills. In graduating with a computing degree it is clear that the institution attests that you have a range of constructive skills. You do not just learn to describe program systems. Rather, you gain considerable experience through the construction of a range of different (representative) systems. The employer expects this experience base in a computing graduate.

In a strong sense, then, the School needs to attest that experimental work is actually carried out by you and that the typical product of experimental work (a submitted assignment report and program) is in fact your work.

In the case where a major submission is simply not the work of the student (unless it is stated clearly that it is a ‘group’ work) then it may well be a plagiarism offence under the Discipline Rules (above). If such an offence is found and proven, then enrolment in the course may be cancelled.

In other cases the submission may be only partly the work of the student. To the extent that it is the work of someone else, and the work submitted does not represent the experience and skill of the student, then it is unacceptable. It cannot be used to attest a student's skills if it is not essentially the work of the student. In this case the assignment will be rejected, in part or in full.

To be specific, this situation commonly becomes evident when two or more assignments are submitted that are strongly similar. By this we mean that it is clear that there is essentially a single piece of work that is being submitted (as independent work) by more than one student. You should be warned that the School has tools that are used to compare all program submissions to a particular assignment and to report programs that are strongly similar. The programs so reported are then carefully checked by the course convenor.

The situation often arises through an excessive degree of collaboration rather than cheating or plagiarism. The following are examples of unacceptable collaboration.

1. Student A is given a copy of work done by student B and copies significant portions of it into his or her own submission perhaps with modifications. The intellectual work is done by student B, yet it is represented by student A as his or her own work.

2. Student C and student D work together and each does half the assignment. They then combine their efforts and each submits a complete assignment as their own work.

3. Student E writes a program with major logical errors. Student F then corrects and rewrites the program to the extent that it can no longer be represented as the work of student E. In fact it probably looks a lot like the submission of student F.

4. Students G and H both receive a great deal of help from an expert J (perhaps J is a college tutor, or is a student who has completed the course) In fact, J essentially writes the assignment for G and H, and their submissions are hence strongly similar.

5. Students K, L and M work as a team throughout all stages of an assignment, regularly inspecting each other's solutions as they evolve. There are few significant differences between the solutions submitted, hence; they cannot be regarded as independent work.

Each of these cases results in submissions that are strongly similar. They are quite easily detected. All of the situations are unacceptable, and all of the submitted work will be rejected, in part or in full. Even in scenario 3, we will reject the work of student F since he has contributed to his work essentially being submitted twice, with different attributions.

On the other hand your computing education will involve many situations of *sensible and acceptable collaboration.*

> In many courses one of the assignments will explicitly state that work will be done in pairs or in teams and all students are expected to contribute. This is called a "group assignment". In this case a joint submission is obviously what we expect.

There are other examples where you can improve your work by working with other people, but you must make sure you do not go too far. Here are some examples of what is acceptable:

> Two or more students discuss an assignment to the extent that the relevant computing principles are identified and an informal design of the program solution is developed. The students then complete the assignment independently and each constructs a corresponding program.
>
> Student G writes a program with logical errors. Student H analyses the program along with student G and helps identify the errors. The *relevant corrections* are then designed and made by student G. Student H submits a *different program written independently* of student G.

In this case the students have each learned the intended lessons and skills of the assignment. Invariably the solution will show the writer's own style and strong similarity will not occur. Indeed such situations will often be established and encouraged by us. In tutorials, groups of students will often be set to collaborate on a problem. In programming laboratories, a demonstrator will typically seek to interact with a small group of students who are all working on the same problem.

But, assignments produced via excessive collaboration cannot be accepted as independent work.

6.5.1 Sensible work practice

You should make sure that other people cannot read your assignment solutions. If your work is copied without your knowledge and submitted by another student then two similar solutions will have been submitted. The situation is identical to a scenario of excessive collaboration, as explained above, and potentially we will reject both submissions. It takes time and effort to establish which the original (genuine) solution is. You should thus be careful with file protections (do not leave your files world or group readable), and you should promptly collect printer output.

You should not give a copy of your assignment program to another student. It could be broadcast further without your knowledge and submitted by a third student. Again it will take time and effort to establish which the original (genuine) solution is. Never email a program 'just to show' someone your version. Similarly, you should not give another student leisure to inspect your solution (for example on a computer screen), as more details than you intended of your own solution may end up in the other student's submission.

Chapter 7: General Help Services

Life at university can sometimes be a bewildering experience. When things go wrong, to whom can you turn? This section attempts to give some pointers on what options are available to students in a broad range of contexts.

Emergencies

From all university phone extensions (not pay phones) you can dial '51111' to reach the switchboard (out of hours this reaches campus security). They will inform the relevant authority.

Counselling

The university's counselling centre is able to assist with a range of issues that may be causing concern to students. They can be contacted on extension '52442', or at the Health and Counselling Building, North Road.

Security

If you require campus security, dial '52249'. Additionally, each laboratory in the CSIT building is fitted with a red 'Emergency Only' button that will alert campus security when pressed. If you are working at night on campus (perhaps in a computing laboratory or the library), campus security is available to escort you safely off campus - call them by phone.

Discrimination and Harassment

The Australian National University is committed to providing a study environment that is safe, fair and free from discrimination and harassment for all members of the University community.

Discrimination is defined as unfair or inequitable treatment on the basis of a person's race, colour, sex, sexual preference or orientation, marital status, pregnancy or potential pregnancy, status as carer, age, disability, ethnic or national origin, breastfeeding requirements, religious or political affiliation, or any other attributes as defined in any legislation that applies to a University activity.

Harassment is defined as behaviour, comments or images, that are unwelcome, offensive or intimidating, and that, in the circumstances, a reasonable person should have expected would be offensive or intimidating.

Under the University's policy for grievance resolution, students can make a complaint by following the procedures that are available on the website for dealing with discrimination or harassment. University-wide contacts can assist in explaining the University policy on grievance resolution and how the process works.

Further information is available on the website at http://students.anu.edu.au/contacts/feedback.php or contact the ANU Students' Association on 02 6125 5849

Study

The Academic Skills and Learning Centre is available to help and advise students experiencing problems with their ability to learn and study effectively. The centre is located on the lower ground floor of the Pauline Griffin Building No 11. The office is open during term and vacations on an appointment basis, the phone number is 02 61252972 and the fax number is 02 61253399. A map is available at http://campusmap.anu.edu.au/displaybldg.asp?no=11

International Student Services

Information for International Students can be found at: http://info.anu.edu.au/studyat/International_Office

Computer Science Students' Association

It is the largest organisation on campus that caters to the needs of IT, Computer Science and Software Engineering (CSIT) students. Information can be found at:
http://cs.club.anu.edu.au/

Welfare

A welfare officer is available for consultation on extension 55849. Consult the ANU web page for other services (e.g., disability) available to students.

First Aid

The current first aid officers in CSIT building #108 are: Elspeth Davies and Bob Edwards. Phone x54043, Room N340, 3rd Floor for assistance.

Transport and Parking

Information about transport and parking can be found at: http://transport.anu.edu.au

University Classifieds

Often text books and other items like furniture can be purchased from the classified section on the University Website. Just go to http://billboard.anu.edu.au/classifieds.asp

Appendix A: Appeals

The University recognises the right of students to seek a review of, and to appeal against, their final result in a course. The ANU Policy is called Assessment Review and Appeals. Here is a useful extract:

STEP 1
Discuss the disputed result with the course convenor.

ACTION TO BE TAKEN
Course convenors will often re-examine a student's work, and will inform the student of their decision.
Many decisions need to go no further than this first step.

STEP 2
If you believe that after Step 1 the result in the course is still inappropriate, discuss the problem with the Delegated Authority.

ACTION TO BE TAKEN
The Delegated Authority will discuss the request with the course convenor.
Apart from determining the rationale for awarding the particular grade, the Delegated Authority will also determine whether established assessment procedures were carried out. At this stage the Delegated Authority may involve a third examiner in the process of reviewing the grade. The Delegated Authority will inform the student of the result of the review process.
Students may also seek the advice of the Dean of Students.

STEP 3
If after Step 2 you still believe the result in the unit is inappropriate, submit to the ANU College Dean, in writing, a formal appeal of the result within 30 working days of the formal notification of results. Reasons why the result is considered inappropriate must be clearly stated and other relevant material included.

STEP 4
The ANU College Dean, after consideration of the students' submission, and following consultation with the course convenor, will determine whether a review panel will be appointed.

Appendix B: Use of University Computing Services

B.1 Information Infrastructure and Service Rules 2012

http://about.anu.edu.au/__documents/rules/infoinfra_and_servrules.pdf

Appendix C: Acceptable Use of Information Technology

https://policies.anu.edu.au/ppl/document/ANUP_000704

Procedure

The University responsibilities in this policy vest in Information Technology Services.

The University, and Heads of Budget Units (where separately managed) must:

1. Investigate non-permitted use of the University's IT and IIS within the bounds of the University Privacy Policy (where the use relates to staff) or the University Discipline Rules 2011 (where the use relates to students) and take appropriate action to protect, preserve, and keep available and accessible, the University's IT and IIS.
2. Ensure users are provided with adequate training and support on the use of IT and the IIS, and the security of information assets.
3. Identify and manage overall risk across the University's information infrastructure.
4. Ensure periodic audits of areas to ensure compliance with relevant policies and procedures.

Heads of Budget Units (where separately managed) must:

5. Apply security management, systems management and network management protection strategies identified in the Infrastructure and Systems Security, Network Operations, and Account Management and Access Procedures.
6. Investigate non-permitted use of the University's information and IIS in accordance with the University's Monitoring and Privacy of Electronic Information Policy.

User Responsibilities:

7. Users of the University's IT and IIS must:

· Use IT and the IIS within the directions, limits and obligations of University Statutes and Rules, and maintain an appropriate level of awareness and compliance with University Policies and Procedures

· Not intentionally connect compromised or unapproved devices or communication equipment to the IIS

· Not intentionally attempt to breach IIS security to access information or other parts of the IIS that is outside their authority

· Not allow access to the IIS to persons outside the University community

· Not use the IIS in a manner that is inconsistent with the provisions of the ANU Code of Conduct and/or Discipline Rules 2011

· Not use another person's credentials, or masquerade as, or represent, another user

· Not use IT or the IIS to harass, threaten, defame, libel, or illegally discriminate, as defined in relevant legislation including the Discrimination and Telecommunications Acts

· Not create, transmit, access, solicit, or knowingly display or store electronic material that is offensive, disrespectful or discriminatory as identified under the ANU Code of Conduct: Respect and fair treatment of people; clauses 18 and 19

· Not contravene any provision of the *Copyright Act 1968*, including, but not limited to, unauthorised use of copyright material, and downloading or sharing pirated content using the University's information infrastructure.

· Maintain an appropriate level of awareness and compliance with University Statutes, Rules, Policies and Procedures governing the use of IT and the IIS

· Use software within the conditions of use specified in the software licence or within any licence agreement between the University and a software vendor

· Not modify or remove University information without authority to do so

· Not breach the confidentiality of others, or the University, and the confidential information of others or the University. Information is considered confidential, whether protected by the computer operating system or not, unless the owner intentionally makes that information available

· Not damage or destroy IT equipment used to access the IIS or any part of the IIS.