Course Outline

Course Code	SENG2011
Course Title	Software Engineering Workshop 2A
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~se2011 (http://cse.unsw.edu.au/~se2011)
Current hand-book entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/SENG2011.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/SENG2011.html)

The timetable for SENG2011 can be found here (/SENG2011/18s2/timetable)

Course Summary

This course teaches practical techniques for computer program development. There are two streams in the course: a *verification stream* that requires students to work individually, and a *project stream* that involves students working in teams.

The verification stream delves into the world of specification, and proving that code is correct, but it does so in a very practical way. In this approach, students will need to think about the behaviour of a system/algorithm separately from the code. In the project stream, mentor-meetings and some lectures will provide direction for teams to build a web application from a team-devised set of requirements. The project is relatively small in scale, owing to time limitations, but it is complex enough to show the importance of collaboration, delegating tasks and project management. Also important is the use of verification, and creativity. Both verification and project streams are heavily tool-based.

Course Aims

- Engineering: learn how to construct an abstract specification of a model of a software system, using a formal tool, and how to marry the implementation to the specification.
- Project Management: to learn the basics of working in a software-project team so that you can deploy effectively the engineering techniques above collaboratively.

Student Learning Outcomes

If you successfully complete this course, you should have acquired a bunch of new skills:

- be able to build a formal specification of the behaviour of a program
- be able to engineer code, not write it in an ad hoc fashion
- be able to reason abstractly about requirements and be able to model them using formal methods
- have developed more skills in project management, and realise the importance of collaboration
- have learned that both behavioural (human based) and technical views (computer based) of design and implementation need to be addressed

Assumed Knowledge

- discrete mathematics, particularly predicate calculus
- programming in a procedural language (e.g. C, Python or Java)
- competence with Unix (e.g. Linux) commands, and interacting with the operating system

Teaching Strategies

SENG2011 is a workshop, which means it is project based and relies less on lectures for information dissemination than a conventional lecture course. While 3 hour lectures are scheduled, in practice the lectures may be shorter than this. The lectures are very important to understanding, particularly of issues in verification. There are no scheduled tutorials or lab sessions, but teams of students are required to meet with an allocated mentor on a regular basis. The mentors fulfil an important role in the course as advisors, and to help students learn the importance of collaboration.

Team meetings of approximately 30 minutes with the mentors should be held weekly. The time and place of the meeting, organised by the mentor, should take into account the availabilities of the mentor and the team members. Members are required to attend meetings in at least 9 of the 11 weeks (starting in week 3, ending week 13). It is recommended that these meetings be held in one of the small seminar rooms in the CSE building, but this will depend on availability. Every meeting should have an agenda that touches on the following points at least: i) a postmortem of the previous week, ii) a plan for coming week, and iii) the delegation of work to members.

Assessment

The total course mark will be the sum of the marks for the following components:

- Assignments (to be done individually):
  - Assignment 1 (10 marks)
  - Assignment 2 (20 marks)
- Project (team work) (35 marks)
  - Initial Project Report
  - Final Project Report
- Examination (35 marks)

The final mark will be capped to 50% in the following circumstances:

- a student fails the exam
- a student fails project management

Project Assessment

The reports will initially be awarded a nominal mark. This mark will be for the report itself, and for the implementation described by the report.

At the time of submission, the mentor will determine the share of each member to the project from the start of the course or from the previous submission, depending on the case. This share takes into consideration:

- the meeting attendance of each member
- the participation of each member in the meetings
- the effort each member has made to the project design, the code, management, and writing the report.

If a mentor determines each member has contributed equally, then each member is awarded the nominal report mark. If a team member has done less than his/her full share, then that individual's project mark will be reduced accordingly.

Late submission

Handing an assignment or project report in late by:

- up to one day (i.e. 24 hours) will be scaled by 85%
- for two days it is 75% (that is total scaling, not in addition to the 85%)
- for three it is 65%
- for four it is 50%
- after four days the submission will not be accepted at all (i.e. is scaled by 0%)

If you think you have sound reasons to request a waiver of these rules, e.g. illness or misadventure, you must submit an official request for special consideration (https://student.unsw.edu.au/special-consideration) , with supporting documentation (e.g. medical certificates) through the formal UNSW central channels (*not* by direct request to the lecturer.)

Supplementary Exam

The document **"Essential Advice for CSE Students" (https://www.engineering.unsw.edu.au/computer-science-engineering/about-us/organisational-structure/student-services/policies/essential-advice-for-cse-students%22)** states the supplementary assessment policy for the school of CSE.

If you are granted a Supplementary examination, then it will be held on the date specified in the above document. If you think that you may be eligible for a supplementary exam, then make sure you are available on that day. It is your responsibility to check at the School Office for details of Supplementary examinations.

Academic Honesty and Plagiarism

Plagiarism is defined as (https://student.unsw.edu.au/what-plagiarism) *using the words or ideas of others and presenting them as your own* . UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Learning Centre: Plagiarism and Academic Integrity (http://www.lc.unsw.edu.au/plagiarism/index.html)
- MyUNSW: Plagiarism (https://student.unsw.edu.au/plagiarism) and Academic Misconduct (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)
- CSE: Addendum to UNSW Plagiarism Guidelines (http://www.cse.unsw.edu.au/~chak/plagiarism/plagiarism.html)
- CSE: Yellow Form (http://www.cse.unsw.edu.au/people/studentoffice/policies/yellowform.html) (whose terms you have agreed to)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism.

Posting questions from the assignments on Q&A websites is considered an attempt to plagiarise, and accordingly such an action will result in penalties.

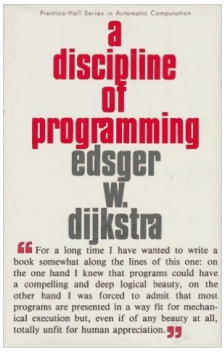
Resources for Students

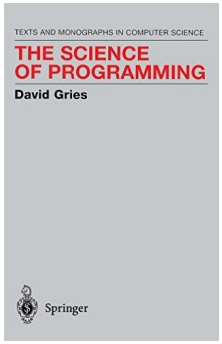
The lecture notes are comprehensive and should be closely studied. Links to supporting documents and articles will also be provided on the website.

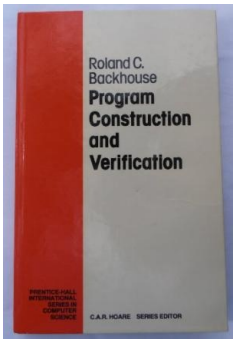
In-depth references that form the backdrop of the course are the following:

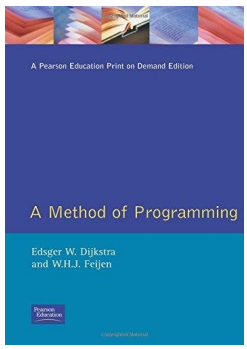
- Edsger W. Dijkstra : **A discipline of programming**
- David Gries : **The science of programming**

- Roland Backhouse: **Program construction and verification**
- Edsger Dijkstra and Wim Feijen : **A method of programming**









These texts provide the theory of verification.

Unfortunately there are no undergraduate level textbooks available on the practice of verification. For this reason, the lecture notes are extensive and detailed. Links to what information there is on the web can be found on the course website.

Course Evaluation and Development

The course was evaluated by students in session 1, 2017. The students were given 7 statements. The responses are shown below, where the percentage that agreed (whether lightly, moderately or strongly) with each statement is shown. The order is from highest to lowest.

The digital resources helped me to learn	96.4%
The amount of assessment was appropriate	94.7%
The assessment tasks helped me to learn	92.9%
Overall I was satisfied with the quality of teaching	92.9%
Overall I was satisfied with the quality of the course	89.4%
I felt part of the learning community	86%
The feedback helped me to learn	70.2%

The best features of this course:

- *A surprising number of students (23 of the 54 respondees) said the project was the best part*
  - (although a few said it was too vaguely formulated)
  - (another few said that Dafny came too late to integrate into the project)
- The second most popular aspect of the course are the lecture notes

The worst features:

- A 3 hour lecture with minimal breaks is too long (~20 students commented on this)
  - a few said lectures are too slow
- Course overlaps with COMP2111 (~10 students comment on this)
- Organisation: need earlier handing out assignments, stagger submission of assignments and project reports, faster marking
- Dafny was too difficult, introduced too late, frustrating and stressful, and students need more help
- No lecture recordings
- Problems with the mentors

Changes in 2018

- Proper breaks will be adhered to, and a faster pace where an alternative is self-study
- All COMP2111-overlapping material (mainly predicate calculus) has been removed
- The project will be worth less marks in 2018 to conform to faculty guidelines; accordingly the scope of the project has been reduced, and there will be more time to address verification in the project
- Dafny will be introduced much earlier, more gradually, with more examples, and more exercises
- Submission dates will be staggered properly. The first assignment will be out very early.
- The LiC will do all the marking, and hopefully do it much faster to improve turnaround
- A new team of mentors has been appointed.
- Lecture recordings should be available in 2018.

Resource created 3 years ago (Sunday 22 July 2018, 10:23:24 PM), last modified 3 years ago (Monday 23 July 2018, 01:21:23 PM).

Comments (/SENG2011/18s2/forums/search?forum_choice=resource/19744) (/SENG2011/18s2/forums/resource/19744)

There are no comments yet.