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)
Handbook 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 (/SENG2011/17s1/timetable) .

SENG1031 is a prerequisite for this course, and COMP2111 is a corequisite.

Course Summary

This course teaches practical techniques for computer program development that help us to proceed from informal but precise *requirements,* via more formal and precise *specifications* through to *implementations* that are correct and easy to understand and maintain. Parallel instruction in the underlying methods used in SENG2011 will come from COMP2111 (http://www.handbook.unsw.edu.au/undergraduate/courses/2014/COMP2111.html) .

In the first part of the course, the lectures, presentations and mentor-meetings will be organized around a sequence of from-requirements-to-program examples that will become increasingly demanding. The problems studied in this part will be relatively small in scale, and assignments will be done individually and by hand: the emphasis is on understanding the techniques both in theory and in practice.

The second part of the course will consist of a groupwork project that will require the application of the techniques studied in the first part to a larger scale problem. This part of the course will use a tool that automates some of the work required to establish that the program developed satisfies its specification. There will also be lectures on Project Management. Students will be expected to apply project management techniques in their project work, and document their application of these techniques.

Course Aims

- Engineering: learn how to construct an abstract specification/model of a software system, applying both informal and formal approaches, and how to take it towards an implementation.
- Project Management: to learn the basics of working in a software-project team so that you can deploy effectively the engineering techniques above while working in collaboration with others.

Student Learning Outcomes

After completing this course, students should:

- be able to build a rigorous specification of a program or program-component's required behaviour ;
- have an appreciation of the role of software engineering in software-system development;
- be able to reason abstractly about requirements and be able to model them using various informal/formal methods; and
- have learned that both behavioural (human based) and technical views (computer based) of design and implementation are important issues.

Assumed Knowledge

- elementary discrete mathematics
- programming at an introductory level in a simple procedural language (e.g. C, Python or similar)
- 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 will be substantially shorter than this. The lectures will be example-driven and are there to provide support in developing practical skills.

There are no scheduled tutorials or lab sessions.

Students will be organised into groups of 5 and are **required** (i.e. it is compulsory) to meet with their assigned group mentor for approximately 30 minutes each week, at a time determined by the mentor, taking into account the availability of members of the group. The group mentors provide a very important role in the course as they are there to support their assigned groups, and the individuals in the groups. It is recommended that meetings with the group mentor be held in one of the small seminar rooms in the CSE building, but this will depend on availability.

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 (45 marks)
  - Deliverables: assuming equal contribution, there will be group marks for 3 of the 4 deliverables:
    - the system: how well has the system been implemented and does it fulfil the requirements?
    - a system report
    - a presentation to the class
    - a project management report outlining the contribution of each member (unmarked)
  - Project management (individual mark): the group mentor will assign a mark to each individual based on the project management report, attendance of meetings, interaction with the mentor, reliability, task completion and communication with other team members.
- Examination (25 marks) (to be done individually of course)

The final mark will be capped to 64% in the following circumstance:

- a student fails the exam

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

- a student has failed project management

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.

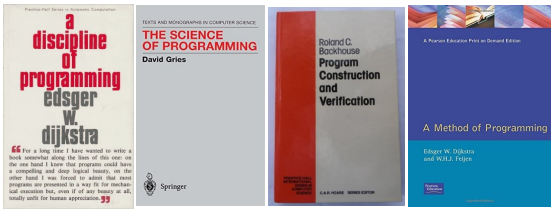
Resources for Students

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 that is studied in the course.

Course Evaluation and Development

The course was evaluated by students in session 1, 2016. The students were asked to agree or disagree with 10 statements about the course. The responses to the statements are shown below, where the percentage that agreed with each statement shown in red font, listed from best to worst.

- The course was effective for developing my thinking skills (e.g. critical analysis, problem solving) **93%**
- The course provided effective opportunities for active student participation in learning activities **89%**
- Lecturer/s handouts are a valuable aid to learning **89%**
- The assessment methods and tasks in this course were appropriate given the course aim **86%**
- The aims of this course were clear to me **79%**
- The course was challenging and interesting **78%**
- In this course the content is organised and presented in a logical and coherent way **68%**
- I was given helpful feedback on how I was going in the course **68%**
- I was provided with clear information about the assessment requirements for this course **65%**

My response to the last 3 points

- *Lecture slides in 16s1 were created on-the-fly causing frequent repairs/improvements*
- *First assignment was very late, other assignments followed too fast, leading to a marking bottleneck*
- *It's hard to integrate verification and project management*

Finally, the students' response to the statement:

- Overall, I was satisfied with the quality of this course **85%**

List of students' comments

The best features of this course were:

- *Ideas have the potential to be interesting*
- *Group Project*
- *Easy.*
- *Yeah,it teach us how to use the Dafny.*
- *Relates to content from COMP2111*
- *The assignment work, as I got to learn Dafny efficiently.*
- *No exam*
- *No final exam probably*
- *The content was fairly easy to understand.*
- *Interesting content*
- *Gives understanding of project management*
- *The practical application of the course content*
- *Group work was challenging and worked well with individual assignments*
- *The concepts taught in the lectures about project management were very useful and important.*
- *Some ideas about handling Dafny were good.*
- *Logic stuff was good. Slides were good. Dafny was good.*
- *Stay having no final exam*

The worst features were:

- *Be a bit more organised.*
- *Less verification*
- *Shorter lectures*
- *Less lengthy lectures.*
- *Be more organised!*
- *A complete revamp is needed. Is it a course about project management or about verification?*
- *Arranging mentor meetings was difficult due to conflicts with other courses*
- *Changing lecturers half way through leaving the course in an unorganized mess*
- *Stop using Dafny and assign assignments earlier than later*
- *Groups of 8 was a big struggle. It was very hard to keep track of where everyone was up to, and at times it was difficult to allocate work.*
- *The 3 hour lecture also deterred many from showing up. Those that did often left during the break. I found it hard to stay focused towards the end of the lecture.*
- *Adding a list of frequently encountered problems students encounter when using Dafny*
- *Not sure if Dafny is suitable for assignments*
- *Spreading course assessments more evenly throughout the semester (instead of the majority at the end).*
- *More organized and not swapping the lecturer mid semester and forcing three assignments and one project in 6 weeks*
- *Project management content is boring and same as last year.*
- *Marking guidelines for project could be clearer.*
- *3 hours is too long for a lecture.*
- *Remove Dafny. I will never use it again and it caused issues for me, causing me to lose marks*

Resource created 5 years ago (Friday 10 February 2017, 04:42:20 PM), last modified 5 years ago (Thursday 02 March 2017, 10:20:15 AM).

Comments (/SENG2011/17s1/forums/search?forum_choice=resource/6236) (/SENG2011/17s1/forums/resource/6236)



Ofir Zeevi (/users/z5117227) 5 years ago (Wed Mar 01 2017 07:59:03 GMT+1100 (AEDT))

Hi Albert, are you able to share an schedule of assessments with us so we can see the weeks our assignments/projects are due? It'll help me plan out my semester