Course Outline Student

Edit Resource (/SENG2011/19T3/resources/32632/edit)

Specification | Make Submission | Check Submission | Collect Submission

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 (/SENG2011/19T3/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 than a conventional lecture course. There is nevertheless a large amount of material available on-line. 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 6 of the 8 weeks** (starting in week 3, ending week 10). The place and time of the meeting is determined by mentor. Every meeting should have an agenda that touches on the following points: 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) (30 marks)
  - Initial Project Report
  - Final Project Report
- Examination (40 marks)

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

- if you fail the exam
- if you fail the project

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 proportionally.

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.

I have no control over who is eligible to take a supplementary exam. This is handled by Student Administration and requires you to apply for Special Consideration.

If you are granted a Supplementary examination, then it will be held on a date specified by the university, or, if it is an on-line examination, specified by the school. 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.

I emphasise:

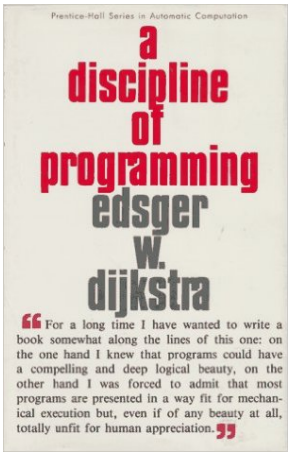
- **Assignments must be done individually.**
- **The project is a group effort.**

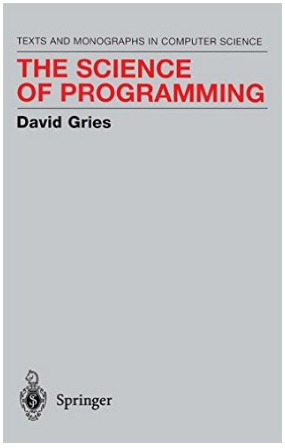
Resources for Students

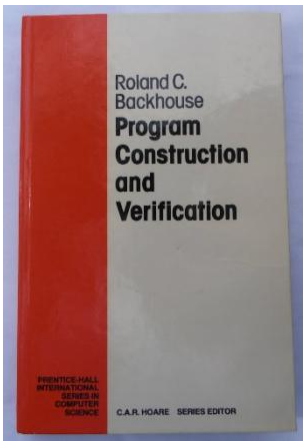
The lecture notes are comprehensive and should be closely studied. Links to supporting documents and articles will also be provided on the website.

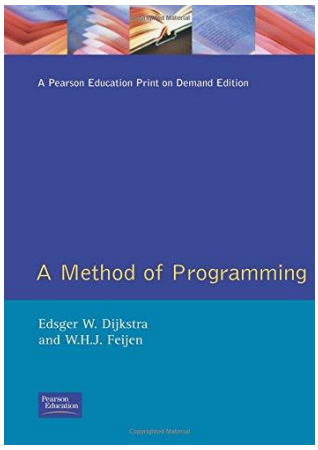
In-depth references that form the backdrop of the course are the following:

- 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 evaluated below consisted of 13 weeks of 3-hour lectures, of course.

The course was evaluated by students in session 2, 2018. The students were given a number of statements. The responses are shown below, where the percentage that agreed (whether lightly, moderately or strongly) with each statement is shown.

I felt part of a learning community	87%
The feedback helped me learn	87%
The digital resources helped me learn	95%
The assessment tasks were appropriate	87%
The amount of assessment was appropriate	90%
Overall I was satisfied with the quality of the teaching	92%
Overall I was satisfied with the quality of the course	79%

The best features of this course:

- Students were split down the middle whether the project or Dafny was the better part of the course.
- A very popular aspect of the course are the lecture notes

The worst features:

- A 3 hour lecture with minimal breaks is too long
- Dafny needs to be introduced sooner
- Marking needs to be faster, less strict, and the report specs clearer
- Assignment 1 was released too late in the session
- The mentors did not know enough about what needed to go into each of the reports

Changes in 2019

- I will try to be more concise in lectures, and more interactive.
- Dafny will be introduced in Week 1, with more examples.
- The first assignment will be out early.
- The more mentors that assist in the marking the faster the turnaround, but the harder it is to ensure uniformity. I need to find a better balance this year.
- There will be a more detailed project spec, and better specs for the interim and final reports for both students and mentors.

Resource created 8 months ago (Sunday 08 September 2019, 12:42:27 PM), last modified 5 months ago (Tuesday 19 November 2019, 05:57:47 PM).